NUMERICAL SCHEMES FOR PARABOLIC PROBLEMS HAVING NON-SMOOTH DATA WITH APPLICATIONS

BY

SALAH ABDO MURSHED ALRABEEI

A Thesis Presented to the

DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

MATHEMATICS

JAN 2017

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **SALAH ABDO MURSHED ALRABEEI** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN MATHEMATICS**.

Thesis Committee

Dr. Mohammad Yousuf (Adviser)

Prof. Abdul Q. M. Khaliq
(Co-adviser)

Prof. Khaled M. Furati
(Member)

Prof. Muhammad A. Bokhari
(Member)

Dr. Faisal A. Fairag (Member)

Dr. Husain Salem Al-Attas
Department Chairman

Dr. Salam A. Zummo
Dean of Graduate Studies

13/1/17
Date

عمادة الدراسات العليا
DEANSHIP OF GRADUATE STUDIES

Dedication

To the greatest man in my life, ***my father.***

To the most affectionate woman in my life, ***my mother.***

To the most beautiful girl in my life, ***my wife****.*

whose sacrificial care for me made it possible for me to complete this work.

This humble work is a sign of my love to you!

ACKNOWLEDGMENTS

After my thanks to Allah, my first acknowledgment goes to my adviser Dr . Mohammad Yousuf, for his continuous help and support. Without his guidance, I would not be able to get such achievement. He also offered me some of his own MATLAB codes for pricing options in Black-Scholes models. I am grateful from the depth to my co- advisor Prof. A. Q. M. Khaliq for many helpful discussions, suggestions, and ideas. He also provided me continuous guidance with the best of his knowledge.

I also wish to thank the other members of my thesis committee Prof. Mohammad Bokhari, Prof. Khaled Furati and Dr. Faisal Fairag for their constructive support encouragement and valuable comments. My thanks and acknowledgments are given to KFUPM for providing me with all needed resources and facilities. .

Many thanks to my families in Yemen and in Saudi Arabia and very special thanks to my beloved mother and father and to my dear wife, who remain willing to engage with the struggle, and ensuing discomfort, of having a partner who is busy with the studies, research. I would like to say a one big ' thank you' to Dr. Khaled Almarbi for all his encouragements and motivations during the whole

time we spent together.

Finally, I would like to thank all my friends Mohammed Aldarowbi , Ahmed Al-Areeq, Ibrahim Althamari, Al-Ezy, thanks for and all my other friends for their support and encouragements.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

PDEs	Partial Differential Equations
PIDEs	Partial Integral Differential Equations
BS	Black-Scholes
JDM	Jump Diffusion Model
ETD	Exponential Time Differencing
RK	Runge-Kutta
ETDRK	Exponential Time Differencing combined with Runge-Kutta
ETDRK4	Exponential Time Differencing combined with Runge-Kutta of fourth order
FD	Finite Difference
FFT	Fast Fourier Transform
BDF2	Backward Differentiation Formula of second order
ETI	Exponential Time Integrator

THESIS ABSTRACT

NAME: Salah Abdo Murshed Alrabeei

TITLE OF STUDY: Numerical Schemes for Parabolic Problems having Non-smooth Data with Applications

MAJOR FIELD: Mathematics

DATE OF DEGREE: Jan 2017

We develop stable, reliable and computationally efficient numerical schemes for solving linear and semi-linear partial integro-differential equations (PIDEs) arising in Financial Mathematics for pricing financial derivatives. These PIDEs are the mathematical formulations of the American and European style options under jump diffusion models.

Non-availability of the analytical solutions for these models motivated us to investigate reliable and efficient numerical schemes that numerically solve these models. Second order central finite difference methods are used for the spatial discretization, and composite Trapezoidal rule to approximate the integral term. We use Padè approximations for the matrix exponential functions to develop time stepping schemes. This achieves second order of convergence in space as well as in

time. To achieve fourth order of convergence in both space and time, we use certain powerful numerical methods such as the Chebyshev Spectral Method for the spatial discretization and the Clenshaw-Curtis Quadrature to approximate the integral term. Then we use the fourth order Padè approximations. Two efficient tools are used in this thesis to acquire an accurate numerical solution in much less computational cost. The FFT algorithm used as a matrix-vector multiplication solver experimentally proves that it is much better than the straightforward matrix-vector multiplication.

Calculating the matrix exponential functions and the inverse of higher order matrices make the schemes computationally expensive. To overcome this issue, we use the partial fraction form of the Padè schemes developed by Saad et.al. [25], and Khaliq et.al. [22] . This reduces the computational cost.

Several Numerical experiments are given to support the analysis and to numerically prove the accuracy and efficiency of the schemes.

ملخص الرسالة

ألاســـــم : صلاح عبده مرشد الربيعي.

عنـــوان الرســـالة : طرق عددية لحل مسائل البارابولك ذات البيانات غير السوية وتطبيقاته

التخصص الرئيسي : ريـاضـــــيات

تاريـــخ الدرجـــة : 2017

في هذه الرسالة نقوم بتطوير طرق عددية تتسم بالاستقرارية والدقة والكفاءة العالية لحل معادلات تفاضلية تكاملية خطية وشبه خطية التي تستخدم في الرياضيات المالية لتسعير عقود الاختيار المالية الاوروبية والامريكية في حال وجود قفزات غير متوقعة في الأسعار.

عدم وجود حلول حقيقية (تحليلية) لهذه المعادلات دفعنا للبحث عن طرق عددية ذات كفاءة لحل هذه المعادلات حلاً تقريباً (عددياً). نستخدم طريقة الفروق المنتهية المركزية من الدرجة الثانية وطريقة شيبيشيف الطيفية(Chebyshiv spectral) ذات التقارب العالي للتقريب المكاني(spatial discretization) وأيضا طريقة شبة المنحرف المركبة وطريقة كلينشاو كورتس التربيعية (Clenshaw-Curtis) لحل التكامل عددياً (تقريبياً) ومن ثم نستخدم تقريب باديه (Pade approximation).

لإيجاد الحل التقريبي للدوال الأسية ذات المصفوفات لتطوير طرق عددية معتمدة على الزمن(time-stepping).
باستخدام هذه الطرق سنضمن أن تكون الحلول العددية ذات تقارب من الرتبة الثانية والرابعة. لتخفيض الزمن المستغرق لتنفيذ هذه الحلول العددية ، نستخدم خوازمية تحويلات فورير السريعة (FFT) لعملية ضرب مصفوفة بمتجة والتي تستهلك وقتا كثيرا عندما نستخدم علمية الضرب الاعتيادي. ليس هذا فقط، وإنما استخدام الشكل الجزي لتقريبات باديه (Pade approximation) في تقريب الدوال الأسية ذات المصفوفات يستهلك جزءً بسيطاً جداً من الوقت بالمقارنة مع حساب قيمة الدالة الأسية ذات المصفوفة المباشرة باستخدام برامج حسابية مثل الماتلاب.

العديد من الأمثلة والتجارب العددية ستُعطى لإظهار وإثبات الكفاءة العالية والدقة والاستقرارية في الطرق العددية المستخدمة.

CHAPTER 1

INTRODUCTION

Parabolic PDEs have smoothing property, that is, the solution is infinitely differentiable for positive time even if the initial data is non-smooth. Non-smooth data occur as a non-smooth initial condition, mismatched boundary and initial conditions or discontinuity in the forcing term. Numerical schemes often develop inaccuracies when solving problems with non-smooth data [16]. Parabolic PDEs with non-smooth data have many applications in various fields such as mechanical engineering, chemistry, and finance. Our interest is to numerically solve a partial integro-differential equation (PIDE) arising in pricing options.

Useful numerical schemes are expected also to have an analogous smoothing property, whereby optimal order convergence is obtained for positive time under data of low regularity. For the homogenous PDE, Khaliq, Twizel, and Voss [26] developed an algorithm for semi-discretized parabolic PDEs using sub-diagonal Padé approximations. Wade *et.al.* [48] used positivity-preserving Padé schemes

for solving the homogenous parabolic PDE with nonsmooth data. For the inhomogeneous PDEs, many authors have contributed in treating the convergence for the cases of non-smooth data using diagonal and sub-diagonal Padè approximations of matrix exponential functions. Wade and Khaliq [47] approximated the exponential function using the rational approximation with three distinct real poles. Voss and Khaliq [46] used rational approximation with four distinct real poles and the partial fraction decomposition to develop a fourth order scheme for inhomogeneous parabolic PDE.

A system of ordinary differential equations (ODEs) is obtained by spatial discretization and Duhamels principle is used to obtain the exact solution of the discretized system. This solution involves matrix exponential functions. Even if the original matrix is sparse, the matrix exponential will not be sparse. The computation of these functions can be a significant amount of work and this will affect the computational efficiency of the scheme. Generally speaking, these calculations of one-dimensional problems may not be very expensive but it will cost too much time for the problem in higher dimensions.

Higher order rational approximations such as Padè approximations have been avoided for long time because they require inverting higher order polynomial of matrices that causes problems due to the ill-conditioning. In this regard there have been some important contributions done by Gallopoulos and Saad [25], Khaliq, Twizell and Voss [26] who used the partial fraction decomposition technique to handle the problems with computational efficiency and accuracy for

implementing higher order schemes. Using this technique, a fourth order scheme can be implemented at the cost of a second order scheme [16, 48].

For the semi-linear parabolic PDEs, Cox and Matthews [20] developed a family of lower and higher order of exponential time differencing Runge–Kutta schemes (ETDRK) for solving nonlinear system of PDEs. However, Kassam and Trefethen [21] showed that the schemes of Cox and Matthews suffer from numerical instability. To address this instability, the later addressed this issue by using complex contour integration, where all eigenvalues must be contained in the a certain contour. Choosing this contour would depend on the discretized matrix. This yields a limitation in Kassam and Trefethen schemes. None of these achievements treated the problems when initial data is non-smooth. Furthermore, Cox and Matthews as well as Kassam and Trefethen modifications require to invert higher order polynomials which causes computational difficulties as well as instability due to the ill-conditioning [41]. Khaliq et. al. [22] introduced a new version of the fourth-order Cox–Matthews, KassamTrefethen ETDRK4 scheme to eliminate these computational difficulties by using Padè approximation to the matrix exponential functions. They also used partial fraction form technique to construct parallel versions of the schemes. In this work, we shall extend the Khaliq et. al [22] schemes to solve partial integro-differential equations (PIDEs) with non-smooth data.

Partial integro-differential equations (PIDEs) arise in many fields such as

physics, engineering and finance. Modeling some real life phenomena, PDEs fails to consider the memory. To treat this, an integral term has been included to the basic partial differential equation which yield a PIDE.

Black and Scholes (BS) [1] proposed a classical option pricing model, modeled by a parabolic PDE which has been widely used for decades in pricing theory. Years later, empirical studies revealed that BS model is inconsistence with the market movement. To overcome this shortcoming, there have been different models such as Jump-diffusion model JDM, with lognormally distribution jumps, proposed by Merton [2], and with double exponentially distribution jumps, purposed by Kou [3]. Both models have finite jump activity and results in a PIDE.

Since there is no closed form solution of such PIDEs, different numerical schemes have been developed by several authors. Operator-splitting technique has been used by Andersen and Andreasen in [4], where Crank-Nicholson scheme is used for the differentiation operator, and explicit time step is used to compute the integral part. Almendral and Oosterlee in [6] used operator-splitting technique combined with fast Fourier transform (FFT) for solving the jump-diffusion model. Multinomial trees method was suggested by Amin [5] but this is restricted by the number of time steps, and it is just of first order accuracy.

Implicit explicit (IMEX) finite difference method was proposed by Cont et al. [7]. Kwon and Lee [11] used IMEX method with three time levels. Recently, IMEX-method has been proposed by Salami and Toivanen [12] to solve the pricing option under jump diffusion model. More recently, IMEX-method combined with

cubic B-spline collocation method has been proposed by Mohan et al [13], this method is of second order accuracy. Exponential time integration (ETI) method is proposed by Tangman, et. al. [9], for solving Merton's jump diffusion model under different options. Lee and Pang [15] avoided the difficulties in computing matrix exponential by using shift-invert Arnoldi approximation.

We intend to extend Khaliq et. al. [22] scheme to solve PIDEs arising in financial mathematics. These PIDEs formulate important models called jump diffusion models. In order to reduce the computational cost we shall use FFT algorithm which works as a matrix-vector multiplication solver . Using this solver we can reduce the complexity from $\mathcal{O}(M^2)$ into $\mathcal{O}(M \log M)$. We shall also construct computationally efficient version of our scheme using partial fraction decomposition technique.

- **Thesis outline:**

The main goal of this thesis is to develop efficient L-stable methods to numerically solve two partial integro-differential equation arising in financial mathematics. The first PIDE is semi-linear for pricing European option under jump diffusion. The second PIDE is for pricing American option under the jump diffusion. Because of the early exercise constraint, the price of an American option has to be at least the same as the payoff function. The purpose of the penalty method is to remove the free boundary by adding a properly chosen penalty term to the

Black–Scholes partial differential equation (PDE) which results in a nonlinear PDE on a fixed rectangular region. Penalty methods force the solution towards a feasible one by penalizing the violations of the early exercise constraint.These L-stable methods are more accurate and computationally efficient. The thesis is organized as follows.

In chapter 1, an introduction to the parabolic PDEs is given. The literature review related to the homogenous and inhomogeneous PDEs is mentioned. Introductions to Padè approximations and partial integro-differential equations is given. The development of the BS models and the JDMs as well as their related work are given.

Chapter 1 is devoted to option pricing theory. Some concepts, notations are given and different types of options are defined. The basic BS models and the Merton's and Kou's JD models are given. The shortcomings of the BS models and how to overcome them by the JDM are illustrated. The full derivations for the BS and JD models and their formulas are given. The American options and how to derive the nonlinear PIDE from the linear complementary problem are mentioned with details.

In chapter 3, the semi-discretization shall be done by the finite difference methods and spectral methods. In case of the finite difference method, the second order central finite difference is given with illustrative example. In Section 3.2 the Chebyshev spectral method is given with some related definitions.

In Chapter 4, Approximating the integral arising in the PIDE requires to use

appropriate methods corresponding to the numerical methods used in the semi-discretization. In Section 4.1, the second order composite Trapezoidal rule is used. Similarly in Section 4.2, the fourth order Clenshaw-Curtis Quadrature method is given. Approximating the integral by this method is used when the Chebyshev spectral method is used to discretize the spatial derivatives.

In chapter 5, the Toeplitz and the circulant matrices are illustrated. Furthermore, some important definitions and concepts related to the Fourier Transforms are given. The Fast Fourier Transform(FFT) and how to use it as an efficient matrix-vector multiplication solver are given in this chapter.

In chapter 6, the definition the Padè approximation and its properties with illustrative example are given in this chapter. In Section 6.1, the partial fraction decomposition form of the Padè schemes and the full derivation of this form are given.

Chapter 7 begins with an introduction and literature review related to the Exponential time differencing (ETD) method. Our motivation to follow the schemes developed by Khaliq et. al [22] is mentioned. In section 7.1, the exact solution for an abstract form of the nonlinear PDE given using Duhamel's principle is mentioned. In Section 7.2, we recall the derivations of the second order and higher order ETD schemes combined with the Runge-Kutta time stepping scheme given by Cox and Matthews [20]. Moreover, Our lower and higher Padè scheme are derived. In Section 7.3, we give the partial fraction decomposition form of Padè schemes given by Khaliq et. al [22]. This chapter is concluded by giving paralleled

algorithms that can be implemented to solve our models using the lower order and the higher order Padè schemes.

In chapter 8, the convergence results are given in this chapter by recalling some theories from previous related works. The stability regions for each scheme is also given by some illustrative figures.

In chapter 9 we experimentally prove the efficiency and accuracy of our two schemes. In seciton 9.1, two experiments are given to confirm the efficiency of the FFT algorithm and the Padè schemes by comparing our methods with some other methods from the literature. In section 9.2 and 9.3, we give several experiments to experimentally confirm the convergence and the stability of our schemes. We test the convergence in space of the two numerical methods used in the semi-discretization. Furthermore, we test the convergence of our lower and higher order schemes in time.

In chapter 10 we conclude our thesis with some remarks and future work are

CHAPTER 2

PRICING OPTIONS

The goal of option theory is to help the investors to manage their money, enhance returns and control their financial future. In this chapter, we first give some terminologies and notations related to option theory and financial mathematics, then we derive some financial models such as Black-Scholes (BS) model and Jump diffusion (JD) models which are considered fundamental models in financial mathematics.

2.1 Terminologies

Before we go through the mathematical models and their formulas, we first need to give some important definitions and notations from option pricing theory that led us to understand the models clearly. In this section we mainly take the definitions and concepts from [56, 57]

Definition 2.1 *An **asset** is a sale object that has a known value at present, but*

it can be changed in the future.

There are a few examples of assets, such as currencies, for example the value of US $1000 in Saudi Riyal, shares in a company and value of gold and oil.

Definition 2.2 *: An* ***option*** *is a sale agreement between two parts, holder and writer, to purchase or to sell, but not the obligation, a particular asset for particular price at particular time in the future.*

There are several types of options, but we are interested in some of them which will be divided into two types depending on the exercising type and exercising time.

Definition 2.3 *:* ***Call option*** *is an option that gives the holder (buyer) the right to buy, but not the obligation, a particular asset for particular price at particular time in the future.*

Definition 2.4 *:* ***Put Option*** *is an option that give the writer (seller) the right to sell, but not the obligation, a particular asset for particular price at particular time in the future.*

Definition 2.5 *:* ***Butterfly Spread Option*** *is a combination of three call options with three exercise prices, in which one contract is bought for two outside exercise prices, E_1 and E_3, and two contracts are sold for the middle exercise price E_2.*

The particular price is called an ***exercise price*** or ***strike price***, the particular time in the future is known as *the* ***expiry date***, or ***maturity*** and the value

of the asset at the maturity is called the ***Payoff*** which is given by the following function

$$\text{Payoff} = \begin{cases} \max(S-E,0) & \text{Call} \\ \max(E-S,0) & \text{Put} \\ \max(S-E_1,0) - 2\max(S-E_2,0) + \max(S-E_3,0) & \text{Butterfly} \end{cases}$$

where E, $E_1, E_2\,, E_3$ are exercise prices and S is the stock price of the options.

Definition 2.6 *:**European option** is an option that can only be exercised (bought or sold) on the expiry date.*

Definition 2.7 *:**American option** is an option that can be exercised (bought or sold) any time to the expiry date.*

Remark 1 *The names European and American options have nothing to do with the geographic regions.*

An Illustrative Example *(European call option)*: The date of today is 18 February 2016, a friend of mine has the right to buy one XYZ share for SR 5000, the exercise price, on 11 October 2016, which is maturity. There will be two possible situations that might happen on the maturity. If the XYZ share price is SR5500 on 11 October 2016, then the man would exercise (buy) the share for only SR5000, and immediately sell it for SR5500, earning SR 500 as a profit. On the other hand, if the asset price is only SR 4500 on the maturity, then the holder would not exercise the option because he would lose SR 500.

Mathematically speaking, let us denote E to the exercise price and T is the maturity, S is the asset, and $S(T)$ denotes the asset price at the maturity. If $S(T) > E$ then the European call option holder might purchase the asset for E and sell it for $S(T)$, earning an amount $S(T) - E$. On the other hand, if $E \geq S(T)$ then the holder may not gain anything.

Similarly, considering the European put option, if $E > S(T)$ then writer might sell the asset for E and buy it from the market for $S(T)$, earning an amount $E - S(T)$ .Whereas, if $S(T) \geq E$ then the holder should not do anything. There are other types of options which depend on the payoff values such as Digital call options, binary options, barrier options, etc.

The following Figures (2.1), (2.2), and (2.3) show the typical plots of the payoff functions of Call, Put and Butterfly options respectively.

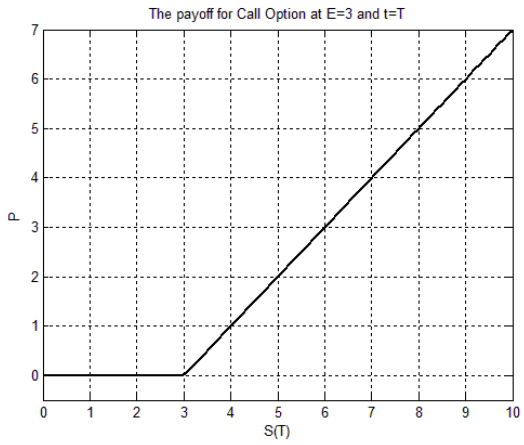


Figure 2.1: payoff function of call Option with strike price E=3.

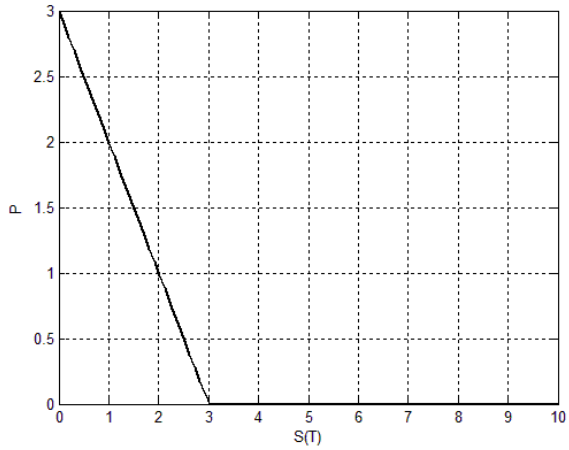


Figure 2.2: payoff function of Put Option with strike price E=3.

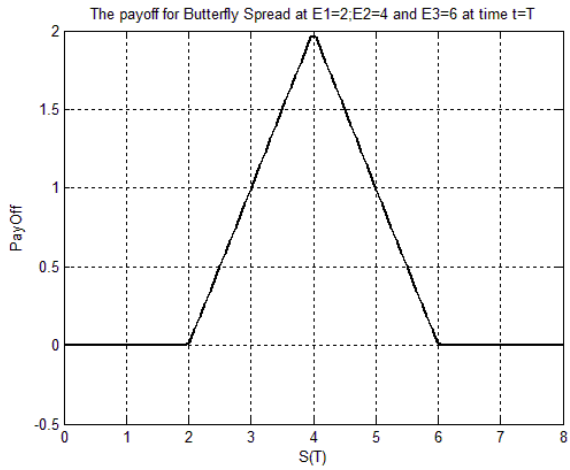


Figure 2.3: payoff function for Butterfly spread with strike prices $E_1 = 2$, $E_2 = 4$, $E_3 = 6$ at the maturity date t=T

2.2 Black-Scholes Model

Black-Scholes (BS) model is one of the fundamental models in pricing options, it is named after Fisher Black and Myron Scholes who developed it in 1973. Thanks to Black-Scholes model, financial mathematics has been developed rapidly. Although BS model was not sufficiently suitable for some realistic assumptions, most of the

other models in option pricing theory are either extensions or generalizations of BS mode [29].

In Black-Scholes model [1], some assumptions have been considered which are given below:

1. Geometric Brownian Motion (GBM) is followed
2. The rate of interest is constant and the underlying stock pays no dividend payments.
3. No transaction costs, no taxes.
4. Short-trade is allowed.
5. Exercising is only at the maturity date

2.2.1 Black-Scholes Model derivation

We denote S to the stock price at time t. Consider the price of the underling asset S is changed by an amount dS during a small time interval dt. Then the value of the underling asset follows the a stochastic differential equation (SDE) given by

$$dS = \mu S dt + \sigma S dW \tag{2.1}$$

where μ is the drift rate, σ is the volatility of the stock, and dW represents the Brownian motion which follows the normal distribution whose mean is 0 and variance is t . Let the value of the option is denoted by $V = V(S, t)$, where V is

twice differentiable w.r.t to S and differentiable w.r.t t . Applying Ito's lemma which is given by

$$dV(S,t) = \left(\frac{\partial V}{\partial t} + \mu S\frac{\partial V}{\partial S} + \frac{\sigma^2}{2}S^2\frac{\partial^2 V}{\partial S^2}\right)dt + \sigma S\frac{\partial V}{\partial S}dW \tag{2.2}$$

We set up a portfolio consisting of one option and $-\frac{\partial V}{\partial S}$ of the underling asset. Then the value Π of portfolio is given by

$$\Pi = V - \frac{\partial V}{\partial S}S \tag{2.3}$$

where the change of the value during dt is given by

$$d\Pi = dV - \frac{\partial V}{\partial S}dS \tag{2.4}$$

Substituting equation (2.1) and (2.2) in equation (2.4) to get

$$d\Pi = \left(\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2\frac{\partial^2 V}{\partial S^2}\right)dt \tag{2.5}$$

Since the portfolio is risk-free, no risk portfolio, then it earns risk-less rate denoted by r. Hence, we get

$$d\Pi = r\Pi dt \tag{2.6}$$

By substituting equation (2.4) and (2.6) in equation (2.5), we get a second order partial differential equation

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV + rS\frac{\partial V}{\partial S} = 0 \tag{2.7}$$

With a suitable boundary and final conditions, equation (2.7) is known by Black-Scholes differential equation.

2.2.2 Black-Scholes Formula

A Black-Scholes partial differential equation (2.7) is of final value problem type. That is, the value of the option is known at the maturity T. Therefore based on Black-Scholes model, the options which restrict the exercising time only at the maturity such as European options have close form solution given by:

$$V(S,t) = \begin{cases} SN(d_1) - Ee^{r(T-t)}N(d_2) & \text{call option} \\ -SN(-d_1) + Ee^{r(T-t)}N(-d_2) & \text{put option} \end{cases}$$

where d_1 and d_2 are given by

$$d_1 = \frac{log(S/E)+(r+\frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}} \ , \ \ d_2 = \frac{log(S/E)+(r-\frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}$$

Whereas, $N(.)$ is the accumulative standard normal distribution function given by:

$$N(y) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{y} e^{-\frac{z^2}{2}} dz$$

The figure (2.4) illustrates the graph of European call option under Black-Scholes model at different maturities

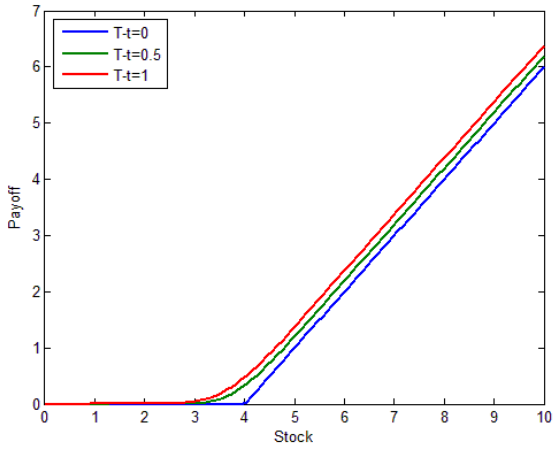


Figure 2.4: Payoff function for European Call Options with strike price $E = 4$, rate of interest $r = 0.1$ and volatility $\sigma = 0.2$ with three different maturity date $T - t = 0$, 0.1 and 1.

2.3 Jump-diffusion Models

Although Black-Scholes Model is considered as the base of option pricing theory, empirical studies revealed that there are shortcomings in BS model when describing the market movements in reality. Several considerable studies have been revealed to overcome these shortcomings. One important approach is what is called jump-diffusion models.

Jump-Diffusion Models (JDMs) is the standard Black-Scholes equation, called diffusion part, added to it a jump component. The diffusion part is modeled by Brownian motion and the jump part is modeled by Poisson process.

2.3.1 Jump-diffusion Model derivation

Consider the equation of underlying process dynamics, proposed by Morten [2]

$$dS = \mu S dt + \sigma S dW + (J-1)S dq \tag{2.8}$$

where the first two parts of the equation are the standard Brownian motion and the last part is the jump, such that J is the size of the positive jump and dq is given by

$$dq = \begin{cases} 0 & \text{with probability } 1 - \lambda dt \\ 1 & \text{with probability } \lambda dt \end{cases}$$

where λ is the Poisson intensity.

Following Lesmana, D. [57], we set up a portfolio consisting of one option with value V and Δ shares at price S. Then the value of portfolio is given by

$$\Pi = V - \Delta S. \tag{2.9}$$

Then the change of the portfolio value is given by

$$d\Pi = d\Pi_1 + d\Pi_2 \tag{2.10}$$

where $d\Pi_1$ and $d\Pi_2$ are the change in the value due to the Brownian part and

jump part respectively. Applying Ito's lemma to get

$$d\Pi = \left[\frac{\partial V}{\partial t} + \mu S\,\frac{\partial V}{\partial S} + \frac{\sigma^2}{2}S^2\frac{\partial^2 V}{\partial S^2} - \Delta\mu S\right]dt + \sigma S\left[\,\frac{\partial V}{\partial S} - \Delta\right]dW \qquad (2.11)$$

The change of the jump of finite size given by

$$d\Pi_2 = \big[V(JS,t) - V(S,t)\big]dq - \Delta(J-1)Sdq \qquad (2.12)$$

we choose $\Delta = \frac{\partial V}{\partial S}$ to hedge the risk of Brownian Motion, thus equation (2.12) can be simplified to

$$d\Pi = \left[\frac{\partial V}{\partial t} + \frac{\sigma^2}{2}S^2\frac{\partial^2 V}{\partial S^2}\right]dt + \big[V(JS,t) - V(S,t)\big]dq - \frac{\partial V}{\partial S}(J-1)Sdq \qquad (2.13)$$

Taking the expectation to the both sides of equation (2.13), to get rid of the random component dq, where the expect value of z is given by

$$E(z) = \int_0^\infty z(J)f(J)dJ \qquad (2.14)$$

where f is a probability density function of the jump size J.

Assuming the probabilities of jump and jump size are not dependent, thus equation (2.14) will be as follows

$$E(d\Pi) = \left[\frac{\partial V}{\partial t} + \frac{\sigma^2}{2}S^2\frac{\partial^2 V}{\partial S^2}\right]dt + E\big[V(JS,t) - V(S,t)\big]\lambda dt \; - \frac{\partial V}{\partial S}E(J-1)S\lambda dt \qquad (2.15)$$

Due to the arbitrage approach, we get

$$E(d\Pi) = r\Pi dt \tag{2.16}$$

i.e the expected return of the portfolio is the risk-free interest rate r.

Putting $\kappa = E(J-1)$, from equations (2.14), (2.15) and (2.16) we get a Partial Integro-differential equation (PIDE) given as

$$\frac{\partial V}{\partial t} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 V}{\partial S^2} + (rS - S\kappa\lambda)\frac{\partial V}{\partial S} - (r-\lambda)V + \lambda \int_0^\infty g(J)V(JS,t)dJ = 0 \tag{2.17}$$

where g is the probability density function, for example in Merton's model g is the log normal density function. The PIDE (2.17) with suitable boundary and final conditions represent the jump diffusion models. For example, a jump diffusion model under European call option is given by

$$\begin{cases} \frac{\partial V}{\partial t} + \frac{\sigma^2}{2} S^2 \frac{\partial^2 V}{\partial S^2} + (rS - S\kappa\lambda)\frac{\partial V}{\partial S} - (r-\lambda)V + \lambda \int_0^\infty g(J)V(JS,t)dJ = 0 \\ V(S,T) = \max(S-E,0) \\ V(0,t) = 0 \\ V(S,t) \cong S - Ee^{-r(T-t)} \quad \text{for large } S \end{cases}$$

2.3.2 Transforming the jump diffusion model PIDE

Unlike Black-Scholes model, jump diffusion models have no close form solutions and need numerical solutions to price the option. Therefore, we need transform the final value problem (2.17) into an initial value problem.

Let $x = log(S)$, for simplicity we assume the exercise price $E = 1$, thus $S = e^x$

Similarly, let $J = e^y \Rightarrow V(S) = V(e^x) = v(x)$. Thus

$$V(JS) = V(e^x e^y) = V(e^{x+y}) = v(x+y)$$

Also the partial derivatives in the differentiation part of the PIDE are transformed to be

$$\frac{\partial v}{\partial x} = S\frac{\partial V}{\partial S}, \quad \frac{\partial^2 v}{\partial x^2} = S^2\frac{\partial^2 V}{\partial S^2} + S\frac{\partial V}{\partial S}$$

Whereas, the integral part is transformed to be

$$\int_0^\infty V(JS,t)g(J)dJ = \int_{-\infty}^\infty v(x+y,t)\phi(y)dy$$

where $\phi(y) = g(e^y)e^y$

We also change the variables by letting $y = z - x$ and $\tau = T - t$. Then the PIDE (2.17) will be given as

$$v_\tau - \frac{\sigma^2}{2}\sigma v_{xx} - (r - \frac{1}{2}\sigma^2 - \kappa\lambda)v_x + (r+\lambda)v - \lambda\int_{-\infty}^\infty v(z,\tau)\phi(z-x)dz = 0 \quad (2.18)$$

With the boundary conditions and payoff functions given, in [12] , as follows

- European call option has the following conditions

$$\begin{cases} v(x,0) = \max(Ee^x - E, 0), & \text{Payoff function} \\ v(x_{\min},\tau) = 0, & \text{lower boundary condition} \\ v(x_{\max},\tau) = Ee^{x_{\max}} - Ee^{-r\tau}, & \text{upper boundary condition} \end{cases}$$

- European put option has the following conditions

$$\begin{cases} v(x,0) = \max(E - Ee^x, 0), & \text{Payoff function} \\ v(x_{\min},\tau) = Ee^{-r\tau}, & \text{lower boundary condition} \\ v(x_{\max},\tau) = 0, & \text{upper boundary condition} \end{cases}$$

- European butterfly spread option has the following conditions

$$\begin{cases} v(x,0) = (E_2e^x - E_1)^+ - 2(E_2e^x - E_2)^+ + (E_2e^x - E_3)^+, & \text{Payoff function} \\ v(x_{\min},\tau) = 0, & \text{lower boundary condition} \\ v(x_{\max},\tau) = 0, & \text{upper boundary condition} \end{cases}$$

where $x_{\min}$ and $x_{\max}$ are the boundaries of the truncated domain Ω.

2.3.3 Merton's Jump Diffusion Model

A first and outstanding jump-diffusion model has been done by Merton [2] to address the shortcomings of Black-Schols model. Merton considered the Brownian

motion follows the normal distribution whose density function is given by

$$\phi(y) = \frac{1}{\sqrt{2\pi}\delta} e^{-\frac{(y-\mu)^2}{2\delta^2}}$$

where μ and δ^2 are the mean and the variance of the distribution respectively. whereas, the expectation of the the impulse function denoted by κ is given by

$$\kappa = Expect(J-1) = e^{(\mu+\frac{\delta^2}{2})} - 1$$

2.3.4 Merton's Jump Diffusion Formula

Although most of the jump diffusion model have no analytical solutions, Merton has given analytical solutions for specific maturity exercising options. (see Morten [2]). For example the analytical solution for European call option under Merton JDM is given by

$$V_C(S,\tau) = \sum_{n=0}^{\infty} \frac{e^{-\acute{\lambda}(\acute{\lambda}\tau)^n}}{n!} V_{BS}(S,\tau,r_n,\sigma_n) \tag{2.19}$$

where V_{BS} is the value of European call option under Black-Scholes model at

$\tau = (T-t)$,

$\acute{\lambda} = \lambda(1+\kappa)$

$\sigma_n = \sqrt{\sigma^2 + n\frac{\delta^2}{\tau}}$,

$\mathrm{r}_n = r - \lambda\kappa + n\frac{log(1+\kappa)}{\tau}$

The graph in figure (2.5) determines price of European call option under

Black_Scholes model and Merton Jump model obtained by the analytical formula in the domain $\Omega = [0, 10]$.

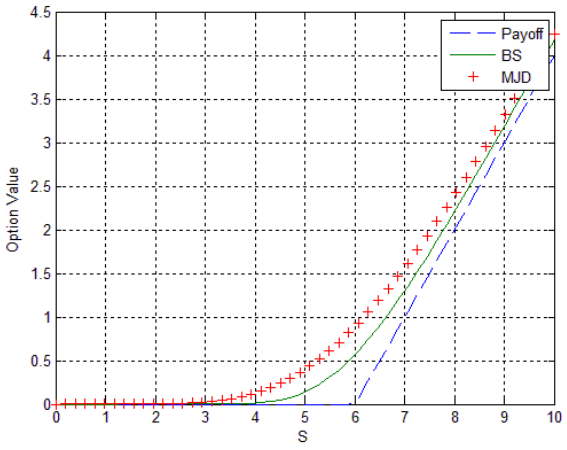


Figure 2.5: Price of the European call option under Black-Scholes model and Merton jump diffusion model at $\lambda = 2$, with $E = 6$, $\sigma = 0.2$,$T = 1$, $r = 0.03$ and $\delta = 0.2$.

2.3.5 Kou's Jump Diffusion Model

Several years later than Merton's contributions [2] in developing Black-Scholes model [1], Steven Kou revealed another model called Kou's jump diffusion model. Kou assumed that the jumps follow a double-exponential distribution [3] whose density function is given by

$$\phi(\xi) = \rho\,\theta_1\,e^{-\theta_1\xi}\,H(\xi) + (1-\rho)\,\theta_2\,e^{\theta_2\xi}\,H(-\xi)$$

where θ_1, θ_2 and ρ are constants providing that $\theta_1 > 1$, $\theta_2 > 0$, $0 < \rho < 1$, and $H(.)$ is the Heaviside function. The expectation of the the impulse function

denoted by κ is given by

$$\kappa = Expect(J-1) = \rho \frac{\theta_1}{\theta_1 - 1} + (1-\rho) \frac{\theta_2}{\theta_2 + 1} - 1$$

All the other assumptions in Merton's model are assumed are considered Kou's model [3].

2.4 American Options

Unlike European options, In the Americans ones the holder can exercise (buy or sell) a certain underling asset for a certain price at any time before the maturity T. Therefore, pricing American options is more challenging than European one because it is required to find the value of the option at each time step and check whether this value is the optimal or not. Mathematically speaking, pricing American option leads to what is called a linear complementarity problem (free boundary value problem) initially proposed by Zvan [17], which is given by

$$\frac{\partial V}{\partial t} + \left(\frac{\sigma^2}{2} S^2 \frac{\partial^2 V}{\partial S^2} + (r - \kappa\lambda) S \frac{\partial V}{\partial S} - (r+\lambda) V + \lambda \int_0^\infty g(J) V(JS, t) dJ \right) \geq 0 \qquad (2.20)$$

$$V - \tilde{V} \geq 0 \qquad (2.21)$$

where $\tilde{V}(S,t)$ is the payoff function, the same function in case of European options, received at the exercising time $t \leq T$ such that one of the equations (2.20) and (2.21) must be strictly equal to zero.

2.4.1 Penalty Method

Since we are not interested in numerically solving a linear complementarity problem (LCP) due to the solving-difficulty, we shall use another method called penalty method. The basic idea of this method is to replace the LCP (2.20) and (2.21) by the partial integro-differential equation (2.17) adding to it an appropriate penalty term resulting a nonlinear PIDE given by

$$\frac{\partial V}{\partial t}+\left(\frac{\sigma^2}{2}S^2\frac{\partial^2 V}{\partial S^2}+(r-\kappa\lambda)S\frac{\partial V}{\partial S}-(r+\lambda)V+\lambda\int_0^\infty g(J)V(JS,t)dJ\right)=\tilde{p}(V,\tilde{V}), \tag{2.22}$$

where $\tilde{p}(V,\tilde{V})$ is the penalty term satisfies

$$\tilde{p}(V,\tilde{V})=\begin{cases} 0 & ,V(S,t)\geq\tilde{V}(S,t) \\ \to\infty & ,V(S,t)<\tilde{V}(S,t) \end{cases}$$

In our work we shall follow Zvan et al. [17] in choosing the penalty term given by

$$p(V,\tilde{V})=\frac{1}{\epsilon}\max\left\{V(S,t)-\tilde{V}(S,t),0\right\},\qquad 0\leq\epsilon\leq 1$$

Like in European options case, we need to transform the PIDE(2.22) into a simpler form given by

$$v_\tau-\frac{\sigma^2}{2}v_{xx}-(r-\frac{1}{2}\sigma^2-\kappa\lambda)v_x+(r+\lambda)v-\lambda\int_{-\infty}^{\infty}v(z,\tau)\phi(z-x)dz=\tilde{p}(v,\tilde{v}) \tag{2.23}$$

with the the following boundary conditions:

- American call option

$$\begin{cases} v(x_{\min}, \tau) = 0, & \text{lower boundary condition} \\ v(x_{\max}, \tau) = Ee^{x_{\max}} - E, & \text{upper boundary condition} \end{cases}$$

- American put option

$$\begin{cases} v(x_{\min}, \tau) = E, & \text{lower boundary condition} \\ v(x_{\max}, \tau) = 0, & \text{upper boundary condition} \end{cases}$$

CHAPTER 3

SPATIAL APPROXIMATION

In this chapter we shall illustrate the numerical techniques that we shall use in this thesis. One Approach for solving a partial differential equation is by setting up a grid or mesh in space and in time, then approximating the solution using a numerical methods such as finite different (FDM), finite element (FEM) or finite volume (FVM). In our work we shall use second order central finite difference method to get second order accuracy in space as well as in time. Furthermore, we shall use spectral methods to get higher order accuracy in space as well as in time.

We first discretize the PIDE in space leading to a system of ordinary differential equations (ODEs) called semi-discretezation to the PIDE, then we chose an appropriate time dependent numerical method to solve this system. These two steps are so called Method of line (MOL) [31].

Unlike the finite element methods and the finite difference methods, the test functions used in the spectral methods are infinitely differentiable global functions.

Whereas, in case of the FE or FD methods the domain is divided into small intervals (elements) and in each interval (element) there is a particular test function, thus the test functions are local functions [32]. Therefore, for the sake of accuracy the spectral methods are more preferable than the finite difference methods.
On the other hand, in the presence of boundary conditions, the use of some spectral methods is less preferable than finite difference methods due to the ill-understood stability problems that may be caused. Furthermore, a time-steps restrictions are sometime required to achieve the desired stability. However, the Chebyshev spectral methods deal with the boundary conditions [33].

3.1 Finite Difference Method

Finite Difference Method is used to solve different types of partial /ordinary differential equations, linear, semi-linear or nonlinear. Briefly, the idea is to transform the derivatives into difference equations over a small interval by using Taylor series expansion. Finite difference methods may approximate the partial derivatives explicitly, implicitly or both.

Lemma 3.1 *Let V(x,t) be a differential function w.r.t x, then the forward finite difference approximation of the partial derivative* $\frac{\partial V}{\partial x}$ *is given by*

$$\frac{\partial V}{\partial x} \cong \frac{V(x+h,t) - V(x,t)}{h} + O(h) \tag{3.1}$$

Lemma 3.2 *Let V(x,t) be a differential function w.r.t t , then the backward finite*

difference approximation of the partial derivative $\frac{\partial V}{\partial t}$ *is given by*

$$\frac{\partial V}{\partial t} \cong \frac{V(x,t) - V(x,t-h)}{h} + O(h) \tag{3.2}$$

Lemma 3.3 *Let V(x,t) be a differential function w.r.t x , then the central finite difference approximation of the partial derivative* $\frac{\partial V}{\partial x}$ *is given by*

$$\frac{\partial V}{\partial x} = \frac{V(x+h,t) - V(x-h,t)}{2h} + O(h^2) \tag{3.3}$$

Lemma 3.4 *Let V(x,t) be a twice differential function w.r.t x , then the central finite difference approximation of the second partial derivative* $\frac{\partial^2 V}{\partial x^2}$ *is given by*

$$\frac{\partial^2 V}{\partial x} \cong \frac{V(x+h,t) - V(x,t) + V(x-h,t)}{h^2} + O(h^2) \tag{3.4}$$

(The proof of the previous lemmas can be obtained using Taylor expansion).

Illustrative Example: Consider

$$\frac{\partial V}{\partial t} + \alpha \frac{\partial^2 V}{\partial x^2} = f(x,t), x \in (0,X)$$

$$V(x,0) = g_1(t), \quad x \in [0,1], \quad V(0,t) = g_2(t), \;\; V(X,t) = g_3(t), \quad t \geq 0 \quad (3.5)$$

where α is constant (for simplicity). We set the spatial mesh by letting its size $h = X/(M+1)$, where M is a positive integer number.

Let $x_m = mh$, where $m = 0, 1, 2..., M+1$. Using lemma (3.4) we replace the

second partial derivative w.r.t x by the second order central finite difference approximation to get

$$\frac{\partial V}{\partial t} + Au = f(t), \quad V(0) = g_1 \tag{3.6}$$

where A is a tri-diagonal MxM matrix ,u, f and g_1 are vectors of length M given by:

$$A = tridiag\left[\frac{\alpha}{h^2}, \frac{-2\alpha}{h^2}, \frac{\alpha}{h^2}\right], \quad f(t) = \begin{vmatrix} -\alpha * g_2(t) + f(x_1, t) \\ f(x_2, t) \\ f(x_3, t) \\ . \\ . \\ -\alpha * g_3(t) + f(x_M, t) \end{vmatrix}, \; g_1(t) = \begin{vmatrix} g_{(x_1)} \\ g_{1(x_2)} \\ g_{(x_3)} \\ . \\ . \\ g_{(x_M)} \end{vmatrix}$$

3.1.1 Spatial Approximation of JDM Using FD Method

Call the partial integro-differential equation(2.18)

$$v_\tau - \frac{1}{2}\sigma^2 v_{xx} - (r - \frac{1}{2}\sigma^2 - \kappa\lambda)v_x + (r + \lambda)v - \lambda \int_{-\infty}^{\infty} v(z, \tau)\phi(z - x)dz = 0$$

We set the spatial mesh in the truncated domain $\Omega = [x_{\min}, x_{\max}]$ by letting its size $h = (x_{\max} - x_{\min})/(M + 1)$, where M is a positive integer number. Set $x_m = x_{\min} + mh$, where $m = 0, 1, 2..., M + 1$. Using lemma (3.3) and (3.4) we replace the first and second partial derivatives w.r.t x by the first and second

order central finite difference approximation respectively to get the following semi-discretization

$$v_\tau + Av = F(v, \tau) \tag{3.7}$$

where A is an $M \times M$ tri-diagonal matrix which is the spatial approximation of the differentiation part of equation(2.18) given by

$$A = \text{tridiag}\left[\frac{c_1}{2h^2} - \frac{c_2}{2h}\ , -\frac{c_1}{h^2} - r - \lambda\ , \frac{c_1}{2h^2} + \frac{c_2}{2h}\right]$$

where

$$c_1 = \sigma^2 \qquad \text{and} \qquad c_2 = r - \lambda\kappa - \frac{\sigma^2}{2}$$

Whereas, $F(v, \tau)$ is the approximation of the integral part which will be discussed in details in the next chapter, $F(v, \tau)$ also contains some other terms depending on the options' boundary conditions.

3.2 Chebyshev Spectral Method

Our aim in this section is to briefly describe the Chebyshev spectral method. The idea of this type of methods is to approximate functions by interpolating polynomials. In this section we mainly follow Trefethen's books (Ch. 5 [35] and Ch. 8 [34]). We introduce the Chebyshev points for a positive integer M

$$x_j = \cos(j\pi/M), \qquad j = 0, 1, 2, ..., M,$$

Following Tangmen [9], we shall use clustered grid nodes such that this clustering will be at the boundaries $\Omega = [x_{\min}, x_{\max}]$ as well as at the singularities E, E_1, E_2 and E_3 which depend on the option. This technique is very effective to reduce the error caused by the non-smoothing data. These clustered grid nodes are given by

$$x = [x_1, x_2]^T \tag{3.8}$$

where

$$x_1(k) = x_{\min} + \left(\frac{E - x_{\min}}{2}\right)\left(1 - \cos\left(\frac{2\pi k}{M}\right)\right), \quad k = 0, 1, ..., \frac{M}{2}\,.$$

$$x_2(l) = E + \left(\frac{x_{\max} - E}{2}\right)\left(1 - \cos\left(\frac{2\pi l}{M}\right)\right), \quad l = 1, ..., \frac{M}{2}\,.$$

For a given function v defined on the Chebyshev points, a discrete derivative w is obtained as follows:

- Interpolate v by a unique polynomial $v_j = p(x_j), \quad 0 \le j \le M$
- Set $w_j = p'(x_j), \quad 0 \le j \le M$

Since the operation is linear, we can represent it in matrix form

$$w = D_M v.$$

where D_M is a square matrix of size $(M+1) \times (M+1)$. Before we give the general formula, we shall find the differentiation matrix at $M = 1$ and $M = 2$, as follows

- $M = 1$: The interpolation points are $x_0 = 1$ and $x_1 = -1$, and the interpolating polynomial written in Lagrange form is given by

$$p(x) = \frac{1}{2}(x+1)v_0 - \frac{1}{2}(x-1)v_1$$

Thus, its derivative is given by

$$p'(x) = \frac{1}{2}v_0 - \frac{1}{2}v_1$$

Therefore, the corresponding matrix form is given by

$$D_1 = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ \\ \frac{1}{2} & -\frac{1}{2} \end{bmatrix}$$

- $M = 2$: The interpolation points are $x_0 = 1$, $x_1 = 0$ and $x_2 = -1$, and the interplant is given by

$$p(x) = \frac{1}{2}x(x+1)v_0 + (1+x)(1-x)v_1 + \frac{1}{2}x(x-1)v_2.$$

where its linear derivative is given by

$$p'(x) = (x + \frac{1}{2})v_0 - 2xv_1 + (x - \frac{1}{2})v_2$$

Therefore, its matrix form is given by

$$D_2 = \begin{bmatrix} \frac{3}{2} & -2 & \frac{1}{2} \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{2} & 2 & -\frac{3}{2} \end{bmatrix}$$

The general Chebyshev differentiation matrix is given in the following theorem

Theorem 3.1 *[34] (Chebyshev differentiation matrix)*

For a nonzero positive integer number M, the first order spectral differentiation matrix D_M has the following entries

$$(D_M)_{00} = \frac{2M^2+1}{6}, \quad (D_M)_{MM} = -\frac{2M^2+1}{6}$$

$$(D_M)_{ij} = \begin{cases} -\frac{x_j}{2(1-x_j^2)}, & i = j, \qquad 1 \le i \le M-1 \\ \frac{c_i}{c_j}\frac{(-1)^{i+j}}{x_i - x_j}, & i \ne j, \quad i,j = 1,2,3,...,M-1 \end{cases}$$

where

$$c_i = \begin{cases} 2, & i = 0 \text{ or } M \\ 1 & \text{Otherwise} \end{cases}$$

Remark 2 *[34] The second order spectral differentiation matrix D_M^2 is the square of the matrix D_M*

3.2.1 Spatial Approximation of the JDM Using Chebyshev Spectral Method

Call the partial integro-differential equation (2.18)

$$v_\tau - \frac{1}{2}\sigma^2 v_{xx} - (r - \frac{1}{2}\sigma^2 - \kappa\lambda)v_x + (r+\lambda)v - \lambda\int_{-\infty}^{\infty} v(z,\tau)\phi(z-x)dz = 0$$

We set the spatial mesh in the truncated domain $\Omega = [x_{\min}, x_{\max}]$ by constructing the clustered nodes (3.8). Then we substitute the first and second partial derivatives in the differentiation part of the PIDE (2.18) by the first and second orders spectral differentiation matrices D_M and D_M^2 respectively. Therefore, we end up with the following semi-discretization

$$v_\tau = Av + F(v,t) \tag{3.9}$$

where

$$A = \frac{1}{2}\sigma^2 D_M^2 + \left(r - \frac{1}{2}\sigma^2 - \kappa\lambda\right)D_M - (r+\lambda)I$$

Whereas, $F(v,\tau)$ is the approximation of the integral part and some other terms depending on the option and its boundary conditions.

CHAPTER 4

INTEGRAL APPROXIMATION

In this chapter we shall briefly discuss how to approximate local integrals as well as non-local integrals. We shall use composite trapezoidal rule to approximate the integrals term arising in JDM models. The advantage of this method is that it gives a special kind of matrices called Toeplize matrix but this method is only of second order accuracy . Therefore, we shall use a higher order method such as quadrature rule method to achieve a fourth order accuracy [27, 28].

4.1 Composite Trapezoidal Rule

The idea of this method in brief is that we divide a given interval $[a, b]$ into subintervals, apply a (simple) trapezoidal rule to each subinterval and finally we sum the results.

Suppose the following integral

$$\int_a^b f(x)dx,$$

for a smooth function $f(x)$.

We partition the interval $[a, b]$ into n-equal subintervals. That is

$$a = x_0 < x_1 < x_2 < ... < x_{n-1} < x_n = b,$$

Where $x_j = a + jh$, and $h = (b - a)/n$, thus the composite trapezoidal rule is

$$\int_a^b f(x)dx = \sum_{j=0}^{n-1} \int_{x_j}^{x_{j+1}} f(x)dx \cong \frac{1}{2} \sum_{j=1}^{n-1} (x_{j+1} - x_j)[f(x_j) + f(x_{j+1})] \tag{4.1}$$

Since h is the width of each subinterval. Thus

$$\int_a^b f(x)dx = h\left[\frac{f(a) + f(b)}{2} + \sum_{j=1}^{n-1} f(x_j)\right]. \tag{4.2}$$

Theorem 4.1.1 *(Composite Trapezoidal Rule Error)*

Let $f''(x)$ be continuous on [a, b]. Let let $I = \int_a^b f(x)dx$.and let G be the approximated integral of $f(x)$ using the trapezoidal rule on [a, b] with an equal partition h, Then there is some $\xi \in$ [a, b] such that

$$I - G = -\frac{(b - a)h^2}{12} f''(\xi) = O(h^2)$$

The proof of this theorem and illustrative examples are available in (Ch 7, [27]).

4.1.1 Approximating The Integral Term Arising in JDMs

The main part of our work is approximating the integral (jump) term arising in jump diffusion models. We call the partial integro-differential equation (2.18)

arising in Merton's model

$$v_\tau - \frac{1}{2}\sigma v_{xx} - (r - \frac{1}{2}\sigma^2 - \kappa\lambda)v_x + (r+\lambda)v - \lambda\int_{-\infty}^{\infty} v(z,\tau)\phi(z-x)dz = 0$$

where ϕ is the normal density function given by

$$\phi(\xi) = \frac{1}{\sqrt{2\pi}\delta}e^{-\frac{(\xi-\mu)^2}{2\delta^2}}$$

where μ and δ^2 are the mean and the variance of the normal distribution respectively. Define $\Omega = [x_{\min}, x_{\max}]$ and denote

$$I(v,x,\tau) = \int_{-\infty}^{\infty} v(z,\tau)\phi(z-x)dz \tag{4.3}$$

We split the integral (4.3) into three integrals as follows,

$$\int_{-\infty}^{\infty} = \int_{\Omega} + \int_{-\infty}^{x_{\min}} + \int_{x_{\max}}^{\infty}$$

The integration over Ω is approximated using composite trapezoidal rule as it is described previously, that is

$$\int_{\Omega} v(z,\tau)\phi(z-x_i)dz = \frac{h}{2}\Big[v(x_{min},\tau)\phi(x_{\min}-x_i) + v(x_{\max},\tau)\phi(x_{\max}-x_i)\Big]$$

$$+h\Big[\sum_{j=1}^{M} v(x_j,\tau)\phi(x_j-x_i)\Big] + O(h^2)$$

By replacing $v(x_{\min},\tau)$ and $v(_{\max},\tau)$ by the boundary conditions corresponding

to the option, we end up with a Toeplitz matrix [19] given by

$$[G]_{j,i} = h\phi(h(i-j)) \tag{4.4}$$

for $i = 0, 1, 2, ...M$, $j = 0, 1, 2, ...M$.

Following [6,11] , the second and the third integrals are either zeros or analytically calculated depending the boundary conditions of the option (call/put/butterfly). For European call option, the second integral is zero because of the integrand $v(x,t) \to 0$ as $x \to -\infty$. Whereas, the third integral is directly calculated and given by,(see Appendix A1, Eq. A.1)),

$$\int_{x_{\max}}^{\infty} v(z,\tau)\phi(z-x_i)dz = Ee^{x_i+\mu+\frac{\delta^2}{2}} f\left(\frac{x_i - x_{\max} + \mu + \delta^2}{\delta}\right) - Ee^{-r\tau} f\left(\frac{x_i - x_{\max} + \mu}{\delta}\right) \tag{4.5}$$

where f is the cumulative normal distribution function given by

$$f(\xi) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\xi} e^{\frac{z^2}{2}} dz\,.$$

Remark 3 *: For the same option under Kou's model , the third integral is analytically computed and given by (see Appendix A1, Eq. A.2)*

$$\int_{x_{\max}}^{\infty} v(z,\tau)\phi(z-x_i)dz = E\rho\theta_1 e^{\theta_1(x_i - x_{\max})}\left(\frac{e^{x_{\max}}}{\theta_1 - 1} - \frac{e^{-rt}}{\theta_1}\right).$$

4.2 Clenshaw-Curtis Quadrature

In this section we briefly determine an important and efficient numerical method for approximating integrals. The general idea of approximating an integral of a function using quadrature rule methods is that we simply sum up a linear combination of the function at certain points. In the Clenshaw-Curtis quadrature, Following Trefethen ([34, 54]), we shall use the fast Fourier transform (FFT) to evaluate the integral of a function $f(x)$ on $[-1, 1]$, where x is in the Chebyshev series. Consider the self-reciprocal function $g(z)$ defined on the $|z| = 1$ by 2-to-1 pointwise equivalence $x = Re(z)$. Let $f(x) \simeq \sum_{i=0}^{n} a_i w_i(x)$ is the interpolant to $f(x)$ in the Chebyshev points $\{x_i\}$, where w_i are the Chebyshev polynomials given by $w_{i+1}(x) = 2xw_i(x) - w_{i-1}(x)$; where $w_0(x) = 1$ and $w_1(x) = x$. Therefore, $f(x)$ corresponding pointwise to the self-reciprocal Laurent polynomial interpolant $f(z) = \frac{1}{2}\sum_{i=0}^{n} a_i(z^i - z^{-i})$ to $g(z)$ in roots of unity $\{z_k\}$.

Thus

$$\int_{-1}^{1} f(x)dx = \int_{-1}^{1} \sum_{i=0}^{n} a_i w_i(x)dx$$

Since $x = Re(z) = \frac{1}{2}(z + z^{-1}) \;\Rightarrow dx/dz = \frac{1}{2}(1 - z^{-2})$ and $w_i(z) = \frac{1}{2}(z^i - z^{-i})$. Therefore,

$$\int_{-1}^{1} f(x)dx = \int_{-1}^{1} \sum_{i=0}^{n} a_i w_i(x)dx = \frac{1}{4}\int_{-1}^{1} \sum_{i=0}^{n} a_i(z^i - z^{-i})(1 - z^{-2})dz \qquad (4.6)$$

$$= \frac{1}{4}\int_{-1}^{1} \sum_{i=0}^{n} a_i(z^i - z^{-i} + z^{i-2} - z^{-2-i})dz = \sum_{\substack{k=0 \\ k \text{ is even}}}^{n} \frac{2a_k}{1 - k^2}$$

Thus, the weights are given by $2/(1-k^2)$ and the nodes $\{a_i\}$ determined by the FFT method are given by [53, 55]

$$a_i = \frac{2}{n}\sum_{k=0}^{n}{}'' f(x_k)\cos\left(\frac{ki\pi}{n}\right)$$

where $x_k = \cos(k\pi/n)$, which is the real part of the FFT. $\sum''$ means that at $k = 0$ and $k = n$ the sum is multiplied by $\frac{1}{2}$.

Remark 4 *Clenshaw-Curtis Quadratures are mostly given in the interval* $[-1, 1]$*, thus to evaluate the integral over [a,b] interval, we need to change the variable as follows*

$$x = \frac{b-a}{2}t + \frac{b+a}{2} \Rightarrow dx = \frac{b-a}{2}dt$$

thus

$$\int_a^b f(x)dx = \int_{-1}^{1} f(\frac{b-a}{2}t + \frac{b+a}{2})\frac{b-a}{2}dt\,.$$

CHAPTER 5

FAST MATRIX-VECTOR PRODUCT SOLVER

In this chapter, an efficient matrix-vector multiplication solver is given for some type of matrices. This solver yields efficient results in much less computational costs. Before we go through the this solver, we first need to give some important definitions and notations related to the work.

Definition 5.1 *A square matrix* $[A]_{M\times M}$ *is called a* ***Toeplitz Matrix*** *if it has the form*

$$A = \begin{bmatrix} a_0 & a_{-1} & \dots & a_{2-M} & a_{1-M} \\ a_1 & a_0 & a_{-1} & \ddots & a_{1-M} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{M-2} & \ddots & a_1 & a_0 & a_{-1} \\ a_{M-1} & a_{M-2} & \dots & a_1 & a_0 \end{bmatrix}$$

Remark 5 *In this thesis, approximating the integral term in the PIDE (2.18) using composite trapezoidal rule gives the toeplitz matrix (4.4).*

Definition 5.2 *A* ***Circulant matrix*** $[C]_{M\times M}$ *is a special case of a Toeplitz matrix when for each of its rows is a right cyclic shift of the row preceding it,*

(i.e)

$$C = \begin{bmatrix} a_0 & a_{M-1} & \dots & a_2 & a_1 \\ a_1 & a_0 & a_{M-1} & \ddots & a_2 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{M-2} & \ddots & a_1 & a_0 & a_{M-1} \\ a_{M-1} & a_{M-2} & \dots & a_1 & a_0 \end{bmatrix}$$

Definition 5.3 *: A* ***Fourier matrix*** $D \in \mathbb{C}^{M\times M}$ *is symmetric and unitary matrix whose components are given as*

$$[D]_{jk} = \frac{e^{-2\pi\hat{i}jk/M}}{\sqrt{M}}, \qquad 0 \le j,k \le M-1 \tag{5.1}$$

where, $\hat{i} = \sqrt{-1}$.

Definition 5.4 *The* ***Discrete Fourier Transform*** *(DFT) is a Transform on* $\mathbb{C}^M$ *given as*

$$[D\{b\}]_j = \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} b_k e^{-2\pi\hat{i}jk/M}, \qquad j = 0,1,...,M-1 \tag{5.2}$$

where b is a vector and D is the Fourier matrix (5.1). Whereas the ***inverse DFT*** *of a vector b is given by*

$$[D^{-1}\{b\}]_j = \frac{1}{\sqrt{M}} \sum_{k=0}^{M-1} b_k e^{2\pi \hat{i} jk/M} = [D * b]_j, \qquad j = 0, 1, ..., M-1 \tag{5.3}$$

5.1 Fast Fourier Transform Algorithm

A Fast Fourier Transform (FFT) algorithm is an efficient technique for multiplying a certain $(M \times M)$ matrix by $(M \times 1)$ vector. This algorithm reduces the computational cost from $O(M^2)$ into $O(M \log M)$ [6, 12, 19].

5.1.1 Evaluating Matrix-vector product by FFT

Consider a square Toeplitz matrix A of size $M \times M$ and a vector b of length M. To evaluate the matrix-vector product $[A]_{M\times M}\ [b]_{M\times 1}$, we first embed the matrix A into a circulant matrix $[C]_{(2M-1)\times(2M-1)}$ and let $\hat{b} = [b, 0, \cdots, 0]$ where $\hat{b}$ is of size $2M-1$. The next step is to apply the DFT to the vector $\hat{b}$ and vector c, where c is the first row and column of the circulant matrix C. Finally apply the inverse DFT transform to the product of transformed two vectors. $(i.e)$

$$\mathbf{v} = \mathbf{ifft}\big(\mathbf{fft}(\mathbf{c}).\mathbf{fft}(\hat{\mathbf{b}})\big) \tag{5.4}$$

The last step is extracting the first M elements of the vector $\mathbf{v}$ to be the desired result.

CHAPTER 6

PADÈ SCHEMES

Computing matrix exponential functions has got a remarkable attention for decades due to its importance in several problems in different fields. Approximating this type of functions not only needs accurate methods, but also needs less time consuming techniques.

Padè approximation, type of rational approximation, is an important tool to approximate matrix exponential functions. It is named after the French mathematician Henri Padé (1863-1953). Padè schemes are the approximants derived by a ratio of two power series approximations. Due to the rational form of Padè approximations, it is better than Taylor expansions when approximating functions containing poles. Following (Thomée [36]), Padé approximation of order $(n+m)$ of the exponential function e^{-x} is given by

$$R_m^n(x) = \frac{P_m^n(x)}{Q_m^n(x)}$$

where $P_m^n(x)$ and $Q_m^n(x)$ are polynomials of order n and m respectively given by

$$P_m^n(x) = \sum_{i=0}^{n} \frac{(m+n+i)n!}{(m+n)!i!(n-i)!}(-x)^i$$

$$Q_m^n(x) = \sum_{j=0}^{m} \frac{(m+n+j)n!}{(m+n)!j!(m-i)!}(x)^j$$

with the property $R_m^n(x) = e^{-x} + O(x^{n+m+1})$ when $x \to 0$.

Examples: The following functions are different lower and higher orders Padé approximations of e^{-x} :

$R_1^0(x) = (1+x)^{-1}$ (Backward Euler)

$R_1^1(x) = (1-\frac{1}{2}x)(1+\frac{1}{2}x)^{-1}$ (Crank-Nicolson)

$R_2^1(x) = (1-\frac{1}{3}x)(1+\frac{2}{3}x+\frac{1}{6}x^2)^{-1}$

$R_2^0(x) = 2(2+2x+x^2)^{-1}$

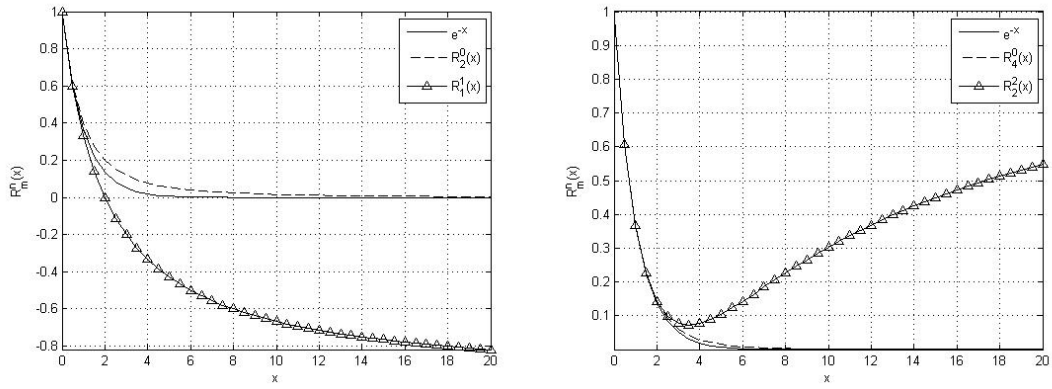
$R_4^0(x) = 24(24+24x+12x^2+4x^3+x^4)^{-1}$

$R_3^3(x) = (1-\frac{1}{2}x+\frac{1}{10}x^2-\frac{1}{120}x^3)(1+\frac{1}{2}x+\frac{1}{10}x^2+\frac{1}{120}x^3)^{-1}$

The following Definitions given in [37] are some properties of rational approximations

Definition 6.1 *A rational approximation $R_m^n(x)$ to the function e^{-x} is said to be* ***A-acceptable*** *if $|R_m^n(x)| < 1$, whenever $\Re(x) < 0$. where $\Re(x)$ is the real part of the complex number x.*

Definition 6.2 *A rational approximation $R_m^n(x)$ to the function e^{-x} is said to be* ***L-acceptable*** *if it is A-acceptable and $|R_m^n(x)| \to 0$ as $\Re(x) \to -\infty$.*

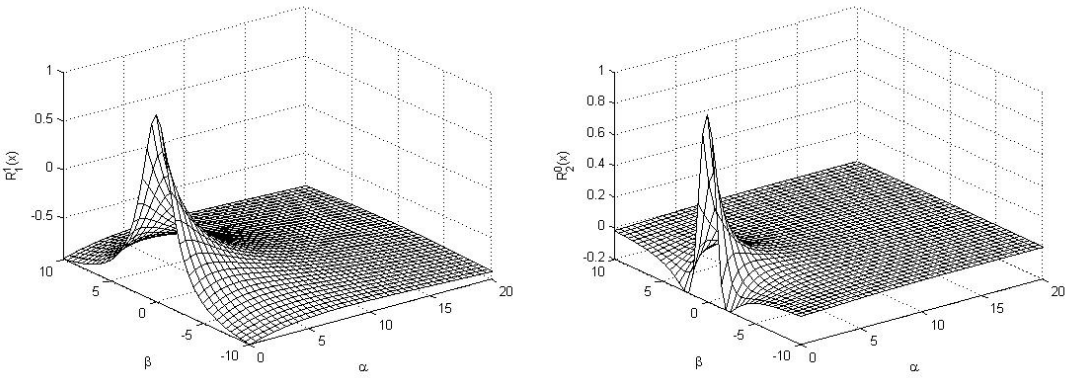


(a) Lower Padè Approximations (b) Higher Padè Approximations

Figure 6.1: Behaviour the functions e^{-x} and lower and higher Padè approximations

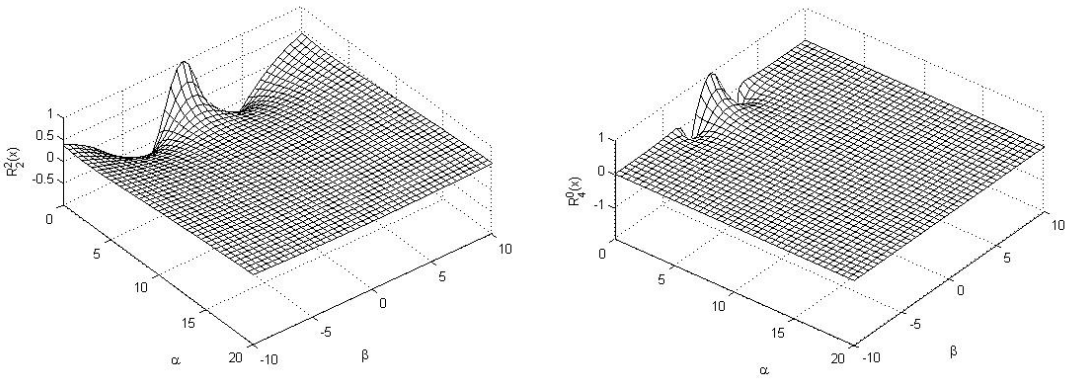
Figure (6.1a) and (6.1b) show the behaviour of the exponential function e^{-x} and some lower Padè approximations such as $R_2^0(x)$ and $R_1^0(x)$, as well as higher Padè approximations such as $R_2^2(x)$ and $R_4^0(x)$.

Whereas the graphs in figures (6.2a)and (6.2b) show the amplifications symbols of some lower and higher Padè approximations for $x = \alpha + i\beta \in [0, 20] \times [-10, 10]$.



(a) (1,1)-Padè (b) (0,2)-Padè

Figure 6.2: Amplification symbols of (1,1)-Padè (left) and (0,2)-Padè (right).



(a) (2,2)-Padè (b) (0,4)-Padè

Figure 6.3: Amplification symbols of (2,2)-Padè (left) and (0,4)-Padè (right).

We can observe that the Padè approximations $R_2^0(x)$ and $R_4^0(x)$ are much better to approximate the exponential function e^{-x} than the others. Therefore, we shall use these two approximants in our work to approximate the exponential function e^{-x} .

6.1 Partial Fraction Of Padè approximations

Approximating the matrix exponential functions using Padè approximations contains quadratic and higher order matrix polynomial in the dominator which causes higher computational complexity and less accuracy. Inverting these polynomial of matrices causes problems due to the ill-conditioning. Moreover, round off errors in computing the powers of matrices can produce bad approximations [38, 39]. In this regards, Gallopoulos and Saad [25], Khaliq, Twizell and Voss [26] have made important contributions to address these difficulties . They used the partial fraction technique to implement higher order Padè scheme. Not only this, but

also they implemented efficient serial and parallel algorithms.

Recall the Padé approximation of the exponential function

$$R_m^n(x) = \frac{P_m^n(x)}{Q_m^n(x)}\,.$$

Follow Khaliq et. al. [26], the partial fraction decomposition of Padè approximations can be divided into two cases.

Case1: $n < m$

The Padé approximation $R_m^n(x)$ is given by

$$R_m^n(x) = \sum_{j=1}^{q_1} \frac{\omega_j}{x - c_j} + 2 \sum_{j=q_1+1}^{q_1+q_2} \mathfrak{R}\Big(\frac{\omega_j}{x - c_j}\Big) \tag{6.1}$$

where q_1 and $2q_2$ are the number of real and non-real roots $\{c_j\}$ of $Q_m^n(x)$, $\mathfrak{R}$ is the real part of a complex component and $\omega_j = P_m^n(c_j)/Q_m^{n\,\prime}(c_j)$.

Case2: $n = m$

$$R_m^n(x) = (-1)^m + \frac{P^{m-1}(x)}{Q_m(x)}\,.$$

Then the Padè approximation is given by

$$R_m^n(x) = (-1)^m + \sum_{j=1}^{q_1} \frac{\omega_j}{x - c_j} + 2 \sum_{j=q_1+1}^{q_1+q_2} \mathfrak{R}\left(\frac{\omega_j}{x - c_j}\right) \tag{6.2}$$

Remark 6 *From now on we denote the Padé approximation $R_m^n(x)$ by $(n, m) - Padé$.*

***Illustrative example**:* For *case1*, $(0,3) - Padé$ given by

$$R_3^0(x) = (1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3)^{-1}$$

To find the poles and weights, we first find the poles of $R_3^0(x)$, roots of $Q_3^0(x)$, to get $c_1 = -1.5960716379833$, $c_2 = -0.7019641810083 - 1.80733949445i$.

To find the weights, we first find derivative of $Q_3^0(x)$ which is given by $Q_3^{0\prime}(x) = 1 + x + \frac{1}{2}x^2$, thus $\omega_1 = 1.475686517795720$, $\omega_2 = -0.7378432588979 + 0.365017840801i$.

Thereof

$$R_3^0(x) = \frac{\omega_1}{x - c_1} + 2\mathfrak{R}\left(\frac{\omega_2}{x - c_2}\right).$$

where

$\omega_1 = 1.475686517795720$, $\omega_2 = -0.7378432588979 + 0.365017840801i$

For *case2* : we choose $(3,3) - Padé$ whose partial fraction form is given by

$$R_3^3(x) = -1 + \frac{\omega_1}{x - c_1} + 2\mathfrak{R}\left(\frac{\omega_2}{x - c_2}\right)$$

where $c_1 = 4.644370709252171$ and $c_2 = 3.6778146453739 + 3.5087619195674i$;

$\omega_1 = -57.20254024714856$, and $\omega_2 = 16.601270123574 + 20.583184279387i$;

Hairer and Wanner [40] described the stability of numerical methods that are used for solving PDEs by the following definitions

Definition 6.3 *A method has an* ***absolutely stable region D*** *if* $|R_m^n(x)| < 1$ *for all* $x \in D$.

Definition 6.4 *A method is called* ***A-stable*** *if* $|R_m^n(x)| < 1$ *for all* x *in the left half-plane.*

Definition 6.5 *A method is called* ***L-stable*** *if it is an A-stable and satisfies*

$$\lim_{x \to -\infty} |R_m^n(x)| = 0\,.$$

CHAPTER 7

EXPONENTIAL TIME DIFFERENCING SCHEMES

Exponential time differencing method (ETD) first appeared in the field of computational electrodynamics [15], but it received much attention when it has been combined with Runge-Kutta time stepping by Cox and Matthews [20]. They developed a family of exponential time differencing Runge-Kutta schemes (ETDRK) for solving nonlinear system of PDEs. However, Kassam and Trefethen [21] showed that Cox and Matthews schemes suffer from numerical instability when the eigenvalues of the discretized matrix A close to zero because of the cancellation errors. Kassam and Trefethen [21] addressed the issue of instability by using complex contour integrations. However, since all eigenvalues must be contained in the a certain contour, choosing this contour depends on the discretized matrix. This yields a limitation in Kassam and Trefethent schemes. Not only this is shortcoming in Cox and Matthews as well as Kassam and Trefethen higher order schemes,

but also they all require to invert matrix higher order polynomials which causes computational difficulties as well as instability due to the ill-conditioning [41]. Furthermore, they all didn't test their schemes for non-smooth initial data which usually causes instability.

Khaliq et. al. [22] overcame all these shortcomings by approximating the exponential matrix by the Padè approximation. They used a partial fraction form of the Padè approximations for solving nonlinear parabolic PDEs with nonsmooth data. As a result, it is only required to solve a simple algebraic system.

In our work, we shall extend the work of Khaliq et. al [22] to solve partial integro-differential equations (PIDEs) with non-smooth data. Particulary, they used their schemes for pricing options under Black-Scholes model (BS) which is a parabolic PDE. Whereas, we shall use ETDRK schemes for pricing options under Merton's and Kou's jump diffusion models.

7.1 The Abstract PDE

Consider a nonlinear initial boundary-value problem (IVP) as follows

$$\begin{cases} v_\tau + Av = F(v,\tau) \quad in\ \mathcal{D}, \qquad \tau \in (0,\bar{\tau}) \\ v = 0 \qquad on\ \partial\mathcal{D}, \qquad \tau \in (0,\bar{t}) \\ v(.,0) = u \qquad\qquad\quad in\ \mathcal{D} \end{cases} \tag{7.1}$$

where $\mathcal{D}$ is a bounded domain in $\mathbb{R}^d$ with Lipschitz boundary and A is a

uniformly elliptic operator and F is a smooth nonlinear on $\mathbb{R}^d$.

We consider the IVP(7.1) is in a Hilbert space $\mathcal{H}$, and consider A to be a linear, self-adjoint, positive definite closed operator with a compact inverse T, defined on a dense domain $D(A) \subset \mathcal{H}$ [22].

We assume that the resolvent set $\phi(A)$ of A satisfies the following:

$$\phi(A) \supset \Sigma_\alpha := \{x \in \mathbb{C} : \alpha < |arg(x)| \leq \pi,\ x \neq 0\}, \qquad \text{for some } \alpha \in (0, \frac{\pi}{2})$$

We also assume there exist $M \geq 1$ such that

$$||(xI - A)^{-1}|| \leq M|x|^{-1}, \quad x \in \Sigma_\alpha$$

It gives that $-A$ is the infinitesimal generator of the analytic semigroup

$$\zeta(\tau) := e^{-\tau A} = \frac{1}{2\pi i} \int_\Gamma e^{-\tau x}(xI - A)^{-1} dx, \tag{7.2}$$

where the contour $\Gamma := \{x \in \mathbb{C} : |arg(x)| = \xi \in (\alpha, \frac{\pi}{2})\}$, with $Im(x)$ decreasing along Γ.

Applying Duhamels principle to equation (7.1) to get

$$v(\tau) = \zeta(\tau)u + \int_0^\tau \zeta(t - s)F(v(s), s)ds, \tag{7.3}$$

For the homogenous case, $F \equiv 0$, the solution operator will only be $\zeta(\tau)u$.

We set $k \geq 0$, $\tau_n = nk$, $0 \leq n \leq N$, replace τ by $\tau + k$ in (7.3), and by the properties the semigroups we obtain

$$v(\tau + k) = \zeta(\tau + k)u + \int_0^{\tau+k} \zeta(\tau + k - s)F(v(s), s)ds$$

$$= \zeta(\tau)\zeta(k)u + \zeta(k)\int_0^{\tau} \zeta(\tau - s)F(v(s), s)ds + \int_{\tau}^{\tau+k} \zeta(\tau + k - s)F(s)ds$$

We change the variables $s - \tau = kr$, to get

$$v(\tau + k) = \zeta(k)v(\tau) + k\int_0^1 \zeta(k - kr)F(v(\tau + kr), \tau + kr)dr \tag{7.4}$$

where the recurrence formula is given by

$$v(\tau_{n+1}) = \zeta(k)v(\tau_n) + k\int_0^1 \zeta(k - kr)F(v(\tau_n + rk), \tau_n + rk)dr \tag{7.5}$$

7.2 Time Stepping Scheme

The difference between a time stepping scheme to another depends on approximating the matrix exponential function and the integral term arising in the recurrence formula (7.5). Cox and Matthews [20] have developed several lower and higher order schemes based on the Runge-Kutta time stepping to approximate the recurrence formula (7.5). Let us denote v_n and F_n to the numerical approximation to $v(\tau_n)$ and $F(v_n, \tau_n)$ respectively. For the simplest approximation to the integral in (7.5), we assume that F is constant between $\tau = \tau_n$ and $\tau = \tau_{n+1}$, that is

$F = F_n + O(k)$, so equation (7.5) becomes **ETD1** first-order accurate given by

$$v_{n+1} = \zeta(k)v_n - (kA)^{-1}(\zeta(k) - I)F_n \tag{7.6}$$

This schemes is used in the field of computational electrodynamics [15]. However, the nonlinear term, F, is not always constat, so we need to use order of accuracy higher than one.

7.2.1 ETDRK2 Scheme

To get a second order approximation to the integral in (7.5) , we use

$$F \simeq F_n + r(F_n - F_{n-1})/k + O(k^2) \tag{7.7}$$

by substituting equation(7.10) in equation (7.5) it follows that

$$v_{n+1} = \zeta(k)v_n + k\int_0^1 \zeta(k - kr)\bigg(F_n + r(F_n - F_{n-1})/k\bigg)dr$$

$$= \zeta(k)v_n + k\int_0^1 \zeta(k - kr)F_n dr + k\int_0^1 r\zeta(k - kr)(F_n - F_{n-1})/k dr$$

$$= \zeta(k)v_n - A^{-1}(\zeta(k) - I)F_n + (-A)^{-2}(\zeta(k) - I + kA)[F_n - F_{n-1}]/k \tag{7.8}$$

Following Cox et al [20], equation(7.8) is the **ETD2** scheme. These type of ETD schemes are of multisteps methods. Unlike ETD methods, RungeKutta (RK) methods is more convenient to use and have small error constants and large stability regions [20]. Therefore, we shall combine the ETD scheme with RK methods

to get Second-Order Runge Kutta ETD Method **ETDRK2** as follows

Let us denote equation(7.6) by

$$a_n = \zeta(k)v_n - A^{-1}(\zeta(k) - I)F_n \tag{7.9}$$

Thus the approximation

$$F = F(v_n, \tau_n) + (\tau - \tau_n)\frac{F(a_n, \tau_n + k) - F(v_n, \tau_n)}{k} + O(k^2) \tag{7.10}$$

is applied and is substituted in equation(7.5) yielding the **ETDRK2** scheme given by

$$v_{n+1} = a_n + (-A)^{-2}(\zeta(k) - I + kA)[F(a_n, \tau_n + k) - F(v_n, \tau_n)]/k \tag{7.11}$$

7.2.2 A second order scheme using (0,2)-Padè

As we mentioned earlier, a serious problem in Cox and Matthews development as well as in Kassam and Trefethen schemes is the difficulty in computing the terms

$$-A^{-1}(\zeta(k) - I) \quad \text{and} \quad (-A)^{-2}(\zeta(k) - I + kA)$$

These two terms contain matrix exponential functions, $\zeta(k) = e^{-kA}$, A^{-1} and A^{-2} need a special treatment to avoid inverting these matrices. Following Khaliq et. al. [22] and Yousuf [24], we use second order (0,2)-Padè approximation of e^{-kA}

arising in equation (7.11) is obtained by

$$v_{n+1} = a_n + (-A)^{-2}(R_2^0(kA) - I + kA)[F(a_n, \tau_n + k) - F(v_n, \tau_n)]/k$$

$$= a_n + k(2I + 2kA + (kA)^2)^{-1}(I + kA)[F(a_n, \tau_{n+1}) - F(v_n, \tau_n)]$$

$$= a_n + k\mathcal{P}_1(kA)[F(a_n, \tau_{n+1}) - F(v_n, \tau_n)] \tag{7.12}$$

where

$$a_n = R_2^0(kA)v_n - A^{-1}(R_2^0(kA) - I)F(v_n, \tau_n) \tag{7.13}$$

$$= R_2^0(kA)v_n + k\mathcal{P}_2(kA)F(v_n, \tau_n) \tag{7.14}$$

and

$$R_2^0(kA) = 2(2I + 2kA + (kA)^2)^{-1}$$

with

$$\mathcal{P}_1(kA) = (I + kA)((2I + 2kA + (kA)^2)^{-1}$$

$$\mathcal{P}_2(kA) = -(kA)^{-1}(R_2^0(kA) - I)$$

7.2.3 ETDRK4 Schemes

The fourth order ETDRK4 Cox's et. al. [20] scheme can be constructed in a similar way to the lower orders. We consider the ETDRK4 scheme given by

$$v_{n+1} = \zeta(k)v_n + \frac{(-A)^{-3}}{k^2}\Big\{F(v_n,\tau_n)\big[-4+kA+\zeta(k)\big(4+3kA+k^2A^2\big)\big]$$

$$+2\big(F(a_n,\tau_n+k/2)+F(b_n,\tau_n+k/2)\big)\big[2-kA+\zeta(k)\big(-2-kA\big)\big]$$

$$+F(c_n,\tau_n+k)\big[-4+3kA-k^2A^2+\zeta(k)\big(4+kA\big)\big]\Big\} \tag{7.15}$$

where

$$a_n = \zeta(k/2)v_n - A^{-1}\big(\zeta(k/2)-I\big)F(v_n,\tau_n) \tag{7.16}$$

$$b_n = \zeta(k/2)v_n - A^{-1}\big(\zeta(k/2)-I\big)F(a_n,\tau_n+k/2) \tag{7.17}$$

$$c_n = \zeta(k/2)a_n - A^{-1}\big(\zeta(k/2)-I\big)(2F(b_n,\tau_n+k/2)-F(v_n,\tau_n) \tag{7.18}$$

These schemes are not only suffer from numerical instability [21], but also it require to compute $-A^{-1}$ and $(-A)^{-3}$ and calculate matrix exponential functions e^{-kA} and $e^{-kA/2}$.

7.2.4 A Fourth order Scheme based on(0,4)-Padè

We follow Khaliq et. al. [22] schemes who developed efficient higher order Padè schemes that don't require inverting matrices. We shall give the notation $R_n^m(kA)$ as (n, m)-Padè approximation of e^{-kA} and $\bar{R}_n^m(kA)$ as (n,m)-Padè approximation of $e^{-kA/2}$. We are interested in the (0,4)-Padè approximation of e^{-kA} to get a

forth order Padè scheme given by

$$v_{n+1} = R_4^0(kA)v_n + \mathcal{P}_1(kA)F(v_n, t_n) + \mathcal{P}_2(kA)\Big(F(\alpha_n, \tau_n + k/2) + F(\beta_n, \tau_n + k/2)\Big) + \mathcal{P}_3(kA)F(\gamma_n, \tau_n) \tag{7.19}$$

where

$$\alpha_n = \bar{R}_4^0(kA)v_n + \bar{\mathcal{P}}(kA)F(v_n, \tau_n)$$

$$\beta_n = \bar{R}_4^0(kA)v_n + \bar{\mathcal{P}}(kA)F(\alpha_n, \tau_n + k/2)$$

$$\gamma_n = \bar{R}_4^0(kA)v_n + \bar{\mathcal{P}}(kA)\Big(2F(\beta_n, \tau_n + k/2) - F(v_n, \tau_n)\Big)$$

with

$$R_4^0(kA) = 24(24I + 24kA + 12k^2A^2 + 4k^3A^3 + k^4A^4)^{-1}$$

$$\mathcal{P}_1(kA) = k(4I - k^2A^2)(24I + 24kA + 12k^2A^2 + 4k^3A^3 + k^4A^4)^{-1}$$

$$\mathcal{P}_2(kA) = k(4I + 2kA + k^2A^2)(24I + 24kA + 12k^2A^2 + 4k^3A^3 + k^4A^4)^{-1}$$

$$\mathcal{P}_3(kA) = k(4I + 4kA + k^2A^2 + k^3A^3)(24I + 24kA + 12k^2A^2 + 4k^3A^3 + k^4A^4)^{-1}$$

$$\bar{R}_4^0(kA) = 384(384I + 192kA + 48k^2A^2 + 8k^3A^3 + k^4A^4)^{-1}$$

$$\bar{\mathcal{P}}(kA) = k(192I + 48kA + 8k^2A^2 + k^3A^3)(384I + 192kA + 48k^2A^2 + 8k^3A^3 + k^4A^4)^{-1}$$

where the rational function $\mathcal{P}_1(kA)$, $\mathcal{P}_2(kA)$ and $\mathcal{P}_3(kA)$ are obtained from the second, third and fourth term of equation (7.15). Whereas, $\bar{\mathcal{P}}(kA)$ is obtained from the second term of equation(7.16).

7.3 Partial Fraction Form Padè Schemes

The ETDRK schemes mentioned in the previous section contain lower and higher order polynomials of matrices which cause computational difficulties. In this regards, Gallopoulos and Saad [25] , Khaliq, Twizell and Voss [26] have made important contributions to address this issue. They used the partial fraction technique to implement the Padè schemes. Not only this, but also they implemented efficient serial and parallel algorithms. Yousuf et al [24] developed algorithms to implement diagonal and damping subdiagonal schemes and obtained the following version of schemes. Although we are interest only on two particular types of Padè approximations, for the sake of generalization, we shall give schemes of (n,m)- Padè in general for any positive integer numbers n , m.

Case1: $n < m$, to compute v_{n+1} we utilize

$$R_m^n(x) = \sum_{j=1}^{q_1} \frac{\omega_j}{x - c_j} + 2 \sum_{j=q_1+1}^{q_1+q_2} \mathfrak{R}\left(\frac{\omega_j}{x - c_j}\right)$$

and

$$\mathcal{P}_i(x) = \sum_{j=1}^{q_1} \frac{\omega_{ij}}{x - c_j} + 2 \sum_{j=q_1+1}^{q_1+q_2} \mathfrak{R}\left(\frac{\omega_{ij}}{x - c_j}\right), \qquad i = 1, 2, 3.$$

where q_1 and $2q_2$ are the number of real and non-real polesc_j of R_m^n and $\mathcal{P}_i$ respectively, with $q_1 + 2q_2 = n$. Whereas, ω_j and ω_{ij} are the weights of R_m^n and $\mathcal{P}_i$ corresponding to the poles c_j.

To compute α_n, β_n and γ_n we utilize

$$\bar{R}_m^n(x) = \sum_{j=1}^{q_1} \frac{\bar{\omega}_j}{x - \bar{c}_j} + 2 \sum_{j=q_1+1}^{q_1+q_2} \mathfrak{R}\left(\frac{\bar{\omega}_j}{x - \bar{c}_j}\right)$$

and

$$\mathcal{P}(x) = \sum_{j=1}^{q_1} \frac{\Omega_j}{x - \bar{c}_j} + 2 \sum_{j=q_1+1}^{q_1+q_2} \mathfrak{R}\left(\frac{\Omega_j}{x - \bar{c}_j}\right)$$

where q_1 and $2q_2$ are the number of real and non-real poles $\bar{c}_j$ of $\bar{R}_m^n$ and $\mathcal{P}$ respectively, where $q_1 + 2q_2 = n$. Whereas, $\bar{\omega}_j$ and Ω_j are the weights of $\bar{R}_m^n$ and $\mathcal{P}$ corresponding to the poles c_j respectively.

Case2: $n = m$, to compute v_{n+1} we utilize

$$R_m^m(x) = (-1)^m + \sum_{j=1}^{q_1} \frac{\omega_j}{x - c_j} + 2 \sum_{j=q_1+1}^{q_1+q_2} \mathfrak{R}\left(\frac{\omega_j}{x - c_j}\right)$$

Whereas, the corresponding $\mathcal{P}_i(x)$, $i = 1, 2, 3.$ are given by the same form in case1. To compute α_n, β_n and γ_n we utilize

$$\bar{R}_m^m(x) = (-1)^m + \sum_{j=1}^{q_1} \frac{\bar{\omega}_j}{x - \bar{c}_j} + 2 \sum_{j=q_1+1}^{q_1+q_2} \mathfrak{R}\left(\frac{\bar{\omega}_j}{x - \bar{c}_j}\right)$$

Whereas, the corresponding $\mathcal{P}(x)$ is given by the form as in case1.

7.4 Algorithms

Although we are interested in (0,2)-Padè for lower order schemes and (0,4)-Padè schemes for higher order schemes, we shall write a general paralleled algorithm for (n,m)-Padè schemes.

For $j = 1$ to $q_1 + q_2$

Step1: **Solve for** $\mathbf{X}_j$

$$\big(kA - \bar{c}_j I\big)X_j = \bar{\omega}_j v_n + k\Omega_j F(v_n, \tau_n)$$

Step2.1: **if n<m, Set**

$$\alpha_n = \sum_{j=1}^{q_1} X_j + 2\sum_{j=q_{1+1}}^{q_1+q_2} \mathfrak{R}(X_j),$$

Step2.2: **if n= m, Set**

$$\alpha_n = (-1)^m v_n + \sum_{j=1}^{q_1} X_j + 2\sum_{j=q_{1+1}}^{q_1+q_2} \mathfrak{R}(X_j)$$

Step3: **Solve for** $\mathbf{Y}_j$

$$\big(kA - \bar{c}_j I\big)Y_j = \bar{\omega}_j v_n + k\Omega_j F(\alpha_n, \tau_n + k/2)$$

Step4.1: **if n<m, Set**

$$\beta_n = \sum_{j=1}^{q_1} Y_j + 2 \sum_{j=q_{1+1}}^{q_1+q_2} \Re(Y_j),$$

Step4.2: **if n= m, Set**

$$\alpha_n = (-1)^m v_n + \sum_{j=1}^{q_1} Y_j + 2 \sum_{j=q_{1+1}}^{q_1+q_2} \Re(Y_j)$$

Step5: **Solve for $\mathbf{Z}_j$**

$$\big(kA - \bar{c}_j I\big) Z_j = \bar{\omega}_j \alpha_n + k\Omega_j \bigg(2F(\beta_n, \tau_n + k/2) - F(v_n, \tau_n) \bigg)$$

Step6.1: **if n<m, Set**

$$\gamma_n = \sum_{j=1}^{q_1} Z_j + 2 \sum_{j=q_{1+1}}^{q_1+q_2} \Re(Z_j),$$

Step6.2: **if n= m, Set**

$$\gamma_n = (-1)^m \alpha_n + \sum_{j=1}^{q_1} Z_j + 2 \sum_{j=q_{1+1}}^{q_1+q_2} \Re(Z_j)$$

Step7: **Solve for $\mathbf{U}_j$**

$$\begin{aligned} \big(kA - c_j I\big) U_j &= \omega_j v_n + k\omega_{1j} F(v_n, \tau_n) + 2k\omega_{2j} \big(F(\alpha_n, \tau_n + k/2) + F(\beta_n, \tau_n + k/2) \big) \\ &+ k\omega_{3j} F(\gamma_n, \tau_n + k) \end{aligned} \tag{7.20}$$

Step8.1: **if n<m, Set**

$$v_{n+1} = \sum_{j=1}^{q_1} U_j + 2 \sum_{j=q_{1+1}}^{q_1+q_2} \Re(U_j),$$

Step8.2: **if n= m, Set**

$$v_{n+1} = (-1)^m v_n + \sum_{j=1}^{q_1} U_j + 2 \sum_{j=q_{1+1}}^{q_1+q_2} \Re(U_j)$$

Step9: **End**

In order to implement the previous paralleled algorithm and use it to solve our models, we first need to compute the poles and weights corresponding to the chosen Padè schemes and the corresponding $\mathcal{P}_j$.

1. For (0,2)-Padè schemes, $R_2^0(x)$ has one non-real pole and its conjugate, so we treat them as one pole, i.e. $q_1 = 0$ and $q_2 = 1$. where $c_1 = -1 + i$, the corresponding weight is $\omega_1 = -i$, and the weights corresponding to the same of pole of the rational functions $\{\mathcal{P}_j\}_{j=1}^2$ are $\omega_{11} = \frac{1}{2}$ and $\omega_{21} = \frac{1}{2} - \frac{1}{2}i$.

2. For (0,4)-Padè schemes, $R_4^0(x)$ has two non-real poles and their conjugates, i.e. $q_1 = 0$ and $q_2 = 2$. where

$$c_1 = -1.72944423106769 + i0.888974376121862$$

$$c_2 = -0.2705557689322 - i2.50477590436244$$

with the corresponding weights

$$\omega_1 = 0.541413348429182 - i1.58885918222330.$$

$$\omega_2 = -0.541413348429154 - i0.248562520866115.$$

where the weights corresponding to the same poles of the rational functions $\{\mathcal{P}_j(x)\}_{j=1}^3$ are

$$\omega_{11} = 0.244153693956274 - i0.0497524711964030.$$

$$\omega_{12} = -0.244153693956268 - i0.0750708534900480.$$

$$\omega_{21} = -0.0240066687966667 - i0.210771761184790.$$

$$\omega_{22} = 0.0240066687966698 + i0.110830774318527.$$

$$\omega_{31} = 0.473042583717175 + i0.293424221840328.$$

$$\omega_{32} = 0.0269574162828241 - i0.165188084403066$$

Whereas, the poles and weights for $\bar{R}_4^0(x)$ are:

$$\bar{c}_1 = -3.45888846213543 - i1.77794875224371.$$

$$\bar{c}_2 = -0.541111537864595 - i5.00955180872487.$$

$$\bar{\omega}_1 = 1.08282669685827 + i3.17771836444659.$$

$$\bar{\omega}_2 = -1.08282669685831 - i0.497125041732246.$$

with the weights corresponding to the same poles of the rational functions $\bar{\mathcal{P}}(x)$ are:

$$\Omega_1 = -0.621169602486758 - i0.599415294095229.$$

$$\Omega_2 = 0.121169602486770 - i0.203064159380992.$$

CHAPTER 8

CONVERGENCE AND STABILITY

In this chapter we shall study the convergence of the ETDRK schemes which have been studied by B. Kleefeld et al [44]. We shall also study the stability analysis of the lower and higher Padè schemes.

8.1 Convergence

Although many schemes that depend on Runge-Kutta time stepping methods such as ETDRK schemes were developed, a complete proof of the convergence has not been achieved [44]. We shall present the convergence results for our interesting L-stable (0,2)-Padè scheme and (0,4)-Padè scheme. Since our focus is to study the convergence of a time-stepping method, we shall consider the error bound between the solutions of the semi-discretized problems (7.9) and (7.11), and the fully discretized problems (7.12) and (7.14).

Remark 7 *: In order to distinguish the semi-discrete problems (7.9) from the the fully discretized problems (7.12) and (7.14) in which the matrix exponential function e^{-kA} is replaced by the Padè approximation $R_2^0(kA)$, we shall denote $\hat{v}$ and $\hat{a}$ to v and a respectively in the semi-discretized problems (7.12) and (7.14).*

Let $\mathcal{H}$ be the finite dimensional subspace of $L^2(\mathcal{D})$, where $\mathcal{D}$ is a bounded domain in $\mathbb{R}^d$. We also assume that $F(t, \hat{v}(t))$ is Lipschitz on $[0,T] \times \mathcal{H}$, that is, it satisfies the following assumption:

Assumption([44]) : *Let $F : [0,T] \times \mathcal{H} \to \mathcal{H}$ and U be an open subset of $[0,T] \times \mathcal{H}$, For every $(t,x) \in U$, there exists a neighbourhood $V \in U$ and a real number L_T such that*

$$||F(t_1,x_1) - F(t_2,x_2)||_{\mathcal{H}} \leq L_T(|t_1 - t_2| + ||x_1 - x_2||_{\mathcal{H}}), \tag{8.1}$$

for every $(t_1,x_2),(t_2,x_2) \in V$.

Theorem 8.1 *([44]): If F is Lipschitz on $[0,T] \times \mathcal{H}$, then for the numerical solution the following error bound holds if $F \in C^2([0,T]; L^1)$,*

$$||\hat{v}(t_n) - v_n||_{\mathcal{H}} \leq C_2 k^2 \max\Bigg(\sup_{0\leq\tau_1\leq T} ||F'(\tau_1, \hat{v}(\tau_1))||_{\mathcal{H}}, \sup_{0\leq\tau_1\leq T} ||F^{(2)}(\tau_1, \hat{v}(\tau_1))||_{\mathcal{H}},$$

$$||\hat{v}_0||_{\mathcal{H}}, ||A\hat{v}_0||_{\mathcal{H}} \Bigg) + C_1 C_2 k^2 + C_1 k^3 \sum_{j=0}^{n-1} ||AF(t_j, \hat{v}_j)||_{\mathcal{H}} \tag{8.2}$$

uniformly on $0 \leq t_n \leq T$. Where C_1 is a constant depends on T but is independent of n, k, and A. Whereas, the constant $C_2 = \max_{t\in[0,T]}\{||F(t,0)||_{\mathcal{H}}\}$ is independent of $\hat{v}$.

The proofs of theorem(8.1) is given by Kleefield ([44], Theorem(4.7)) where the (1,1)-Padè approximation is used to approximate the exponential of matrices. Whereas, we approximate the exponential functions by the $(0,2) - Padè$ and $(0,4) - Padè$ approximations.

8.2 Stability Analysis

In this section we shall study the stability conditions of the $(0,2) - Padè$ and $(0,4) - Padè$ schemes, (see [20, 39]). Consider the nonlinear ODE,

$$v_t = cv + F(v) \tag{8.3}$$

where $F(v)$ is the nonlinear term. We assume that there exists a fixed point $v_0 = v(t_0)$, such that $cv_0 + F(v_0) = 0$. We linearize about the fixed point to lead to

$$v_t = cv + \lambda v. \tag{8.4}$$

where v becomes the perturbation to v_0, whereas, $\lambda = F'(v_0)$.

Following Cox et. al in [20], if $\Re(c + \lambda) < 0$, then the fixed point v_0 is stable.

To obtain the stability region of the numerical methods, we first denote $\xi = \lambda k$ and $\eta = ck$, where k is the time step-size, then we apply equation (7.12) or equation (7.19), in case of the second order or fourth order L-stable method respectively, to the ODE (8.4) leading to a recurrence relation involving v_n and

v_{n+1}. The following amplification factor corresponding to the (0,2)-Padè scheme can be computed by any mathematical software.

$$\frac{v_{n+1}}{v_n} = r(\xi, \eta) = \frac{\xi^2\eta^2 - 3\xi^2\eta + 2\xi^2 + \xi\eta^2 - 4\xi\eta + 4\xi + 2\eta^2 - 4\eta + 4}{(\eta^2 - 2\eta + 2)^2} \tag{8.5}$$

Generally speaking, the parameters c and λ are complex and ξ and η as well. Therefore, the the stability region of the (0,2)-Padè scheme is four dimensional, which make it difficult to plot [20]. Hence different approaches have been used to overcome this issue. Cox et al. [20] set both ξ and η as real, whereas, Beylkin [45] assumed that ξ is complex and η is fixed and real.

According to Beylkin [45], for a better useful method, the stability regions grow as $|\eta|$ becomes larger. Therefore, We shall fix η with several negative real values , $\eta = 0$, $\eta = -5$, $\eta = -10$ and $\eta = -20$, in the complex ξ-plane.

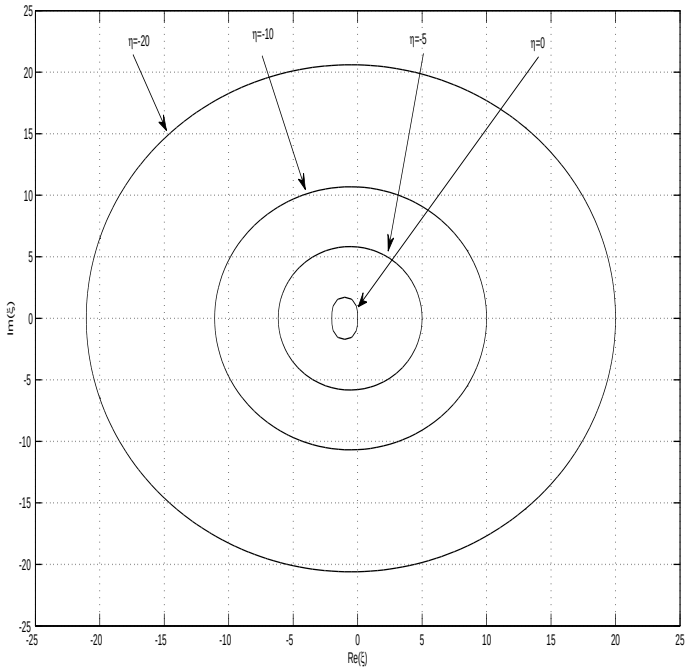


Figure 8.1: Stability regions(inside the circles) of (0,2)-Padè scheme in the complex ξ-plane

We can observe from figure (8.1) that the stability region tends to the second order Runge-Kutta scheme as $\eta \to 0$, and it grows as η decreases from -10 to -20. This result gives an indication of the stability of the (0,2)-Padè scheme.

Similarly, to present the stability regions of the higher order L-stable method, we shall first obtain the amplification factor of the fourth order L-stable (0,4)-Padè scheme. We follow the same procedure used by in [20, 45] to derive the amplification factor corresponding to the (0,4)-Padè scheme. Using Matlab©2014b, we end up with the following amplification factor

$$\frac{v_{n+1}}{v_n} = r(\xi, \eta) = \frac{c_0 + c_1\xi + c_2\xi^2 + c_3\xi^3 + c_4\xi^4}{D} \tag{8.6}$$

where

$D = (24 - 24\eta + 12\eta^2 - 4\eta^3 + \eta^4)(384 - 192\eta + 48\eta^2 - 8\eta^3 + \eta^4)^3$

$c_1 = 905969664 + 1472200704\eta + 1160773632\eta^2 + 608698368\eta^3 + 229441536\eta^4 + 63111168\eta^5 + 12238848\eta^6 + 1376256\eta^7 - 61440\eta^8 - 69376\eta^9 - 17856\eta^{10} - 2912\eta^{11} - 332\eta^{12} - 24\eta^{13} - \eta^{14}$

$c_2 = 226492416 - 283115520\eta + 179306496\eta^2 - 64880640\eta^3 + 12779520\eta^5 - 7643136\eta^6 + 2810880\eta^7 - 762624\eta^8 + 159616\eta^9 - 26400\eta^{10} + 3440\eta^{11} - 338\eta^{12} + 24\eta^{13} - \eta^{14}$;

$c3 = 113246208 - 169869312\eta + 122683392\eta^2 - 70778880\eta^3 + 26738688\eta^4 - 7495680\eta^5 + 1665024\eta^6 - 287744\eta^7 + 40384\eta^8 - 4544\eta^9 + 388\eta^{10} - 26\eta^{11} + \eta^{12}$;

$c4 = 28311552 - 49545216\eta + 37158912\eta^2 - 23887872\eta^3 + 10764288\eta^4 - 3566592\eta^5 + 925952\eta^6 - 187712\eta^7 + 30176\eta^8 - 3828\eta^9 + 364\eta^{10} - 25\eta^{11} + \eta^{12}$;

In figure (8.2), we show the stability region in the complex ξ-plane for different negative real η, $\eta = 0$, $\eta = -5$, $\eta = -10$ and $\eta = -20$. When $\eta \to 0$ the stability region tends to the fourth-order Runge-Kutta scheme; and, as η decreases from -10 to -20 the region grows. This region gives an indication of the stability of the (0,4)-Padè.

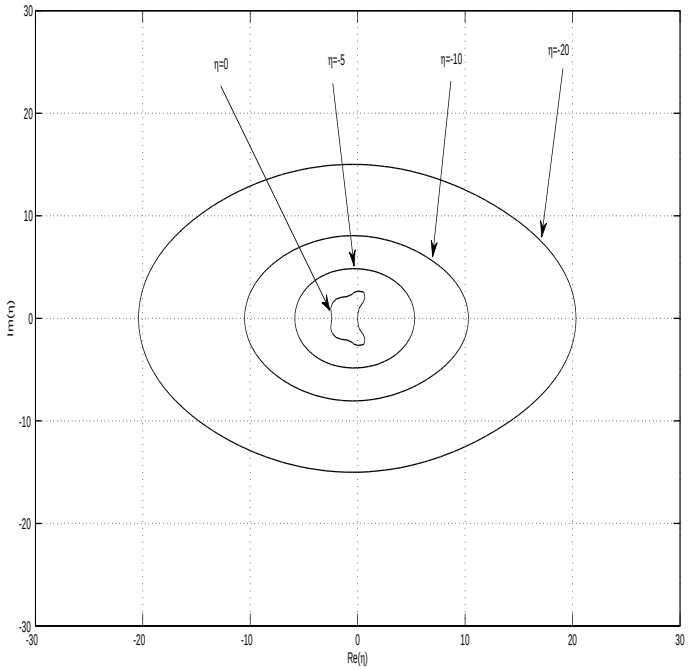


Figure 8.2: Stability regions(inside the circles) of (0,4)-Padè scheme in the complex ξ-plane

CHAPTER 9

NUMERICAL EXPERIMENTS

In this chapter we shall demonstrate the performance of our schemes by showing their efficiency and accuracy. Several techniques are used to overcome some difficulties in the computations. Our concern is not only the convergence, but also on the computational cost, thus our experiments are divided into two approaches, the first approach is to show efficiency of our schemes by comparing the computational cost of our schemes and of others from the literature. The second approach is to show the effectiveness of our schemes by achieving the desired order of convergence in space as well as in time.

Although some numerical schemes look accurate and stable when approximating the price of such option, they lack these properties when approximating the sensitivities of the options known as Greeks [13]. Therefore, we shall graphically show the stability of some of the Greeks (Delta Δ and Γ) where Delta measures the rate of change of option value with respect to changes in the underlying asset's price. Whereas, Gamma represents the rate of change in the delta with respect

to changes in the underlying price. Mathematically speaking, Delta and Gamma are the first and second derivatives of the numerical solution of the priced option respectively [13].

$$\Delta = \frac{\partial V}{\partial S} \qquad \text{and} \qquad \Gamma = \frac{\partial^2 V}{\partial S^2}$$

All the experiments are taken from well known literature and have important practical significance. We have used our schemes to evaluate the price of European and American call, put and butterfly options under Merton's [2] and Kou's [3] jump diffusion models. All the numerical experiments results were computed using Matlab on PC running with processer core i3.

9.1 Computational Costs

In this section we have two experiments, the first is used to show the efficiency of the FFT algorithm as a matrix-vector multiplication solver. The second experiment is to show the efficiency of the partial fraction form of the Padè scheme.

9.1.1 FFT Algorithm Efficiency

To compare the efficiency of the FFT algorithm when it is used as a matrix-vector multiplication solver, we compare the CPU time when using the (0,4)-Padè scheme with and without the FFT algorithm . Our schemes is used to solve the European call option under Kou's JDM in the truncated domain $\Omega = [-6, 6]$, with the parameters from the literature [6] shown in Table(9.1)

Table 9.1: Parameters of Kou's model

E	σ	ρ	r	λ	α_1	α_2	T
1	0.2	0.5	0	0.2	3	2	0.2

Table 9.2: FFT algorithm vs the straightforward multiplication.

		Straightforward Multiplication	FFT Algorithm
M	N	CPU(seconds)	CPU(seconds)
257	40	0.250	0.277
513	80	0.939	0.822
1025	160	4.777	4.169
2049	320	27.683	22.018
4097	640	176.423	130.603
8193	1280	1265.331	740.895

From table (9.2) we can observe that for small systems of equations with a few iterations, FFT algorithm is not that useful. However, in the bigger systems and many iterations it becomes very useful, it reduces the computational cost about half of the straightforward multiplication.

9.1.2 Padè Schemes Efficiency

In this experiment we consider the numerical solution for European call option under Merton's jump obtained by (0,2)-Padè scheme and exponential Time Integrator (ETI) method used by [42] , under the data set given in Table (9.3) and

the truncated domain $x_{\min} = -2$ and $x_{\max} = 2$.

In the ETI method , we follow [42] by computing the matrix exponential function e^{-Ax} explicitly using the built-in function *expm* in Matlab© and we also compute the inverse of the discretized matrix A by the built-in function *inv*

Table 9.3: Parameters and notations of the 2nd experiment.

Exercise Price	Volatility	Variance	interest Rate	jump size	Mean	Maturity
E	σ	δ	r	λ	μ	T
100	0.3	0.5	0	1	0	0.5

In this experiment we use the Chebychev spectral method in the spatial discret ization yielding a dense matrix A, thus it causes difficulty in the computation which makes the need for the Padè scheme in the partial fraction decomposition necessary.

The results given in table (9.4) are obtained by calculating the price of European call option under Merton's JDM at the asset price $S = E$ at each refinement, the number of space-step (M) is taken as equal to the number of time-step (N). The error is calculated by

$$Error = |V_{approx}(E) - V_{exact}(E)| \tag{9.1}$$

where the exact solution is computed by the analytical formula [2] given in (Chapter 2).

Table 9.4: L–Stable scheme vs ETI scheme

		L–Stable Scheme		ETI	
M	N	Error	CPU	Error	CPU
40	40	2.706e -02	0.03714	1.929e-02	0.27055
80	80	3.180 e-03	0.23670	1.216e-03	0.48838
160	160	5.778 e-05	0.39437	8.343 e-06	2.72692
320	320	1.365e-05	1.50878	1.260e-6	48.91743
640	640	3.918e-06	10.36289	8.074e-7	683.06578

We can easily notice from Table(9.4) that the accuracy of the solution is almost the same in the both methods. However, the Padè scheme is extremely faster than the ETI scheme especially, when the time-steps is small as well as the size of the matrix A is larger.

9.2 Pricing European Options

In this section we shall test the effectiveness of our schemes by evaluating the price of European call and put under Merton's and Kou's JDM. Our main target here is to experimentally show that our second and fourth order schemes achieve the convergence order.

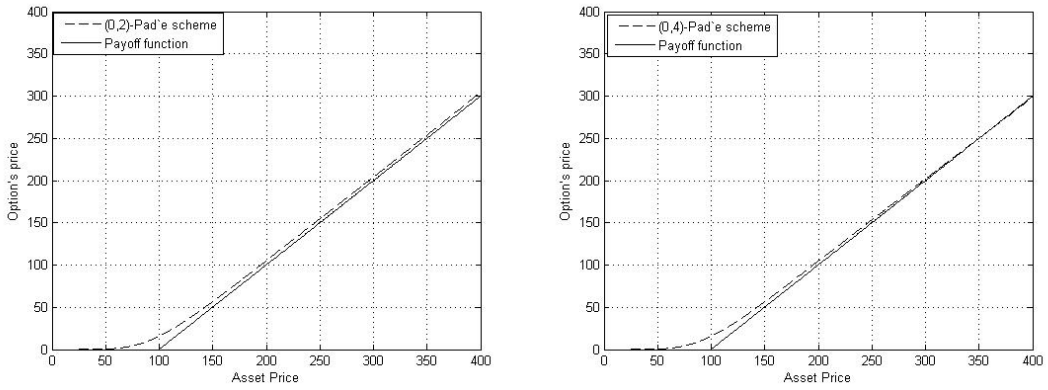
9.2.1 European Call Option Under Merton's JDM

We consider the numerical solution for European call options under Merton's JDM obtained by the (0,2)-Padè scheme as well as the (0,4)-Padè scheme in the truncated domain $x_{\min} = -1.5$ and $x_{\max} = 1.5$ with the parameters in table (9.5) taken from the literature [43]

Table 9.5: Parameters of the 3nd experiment.

E	σ	δ	r	λ	μ	T
100	0.2	0.2	0.05	2	0	0.5

The graphs in figure (9.8a) and (9.1b) show the behavior of the numerical solutions of the European call option obtained by the (0,2)-Padè scheme as well as the (0,4)-Padè scheme.

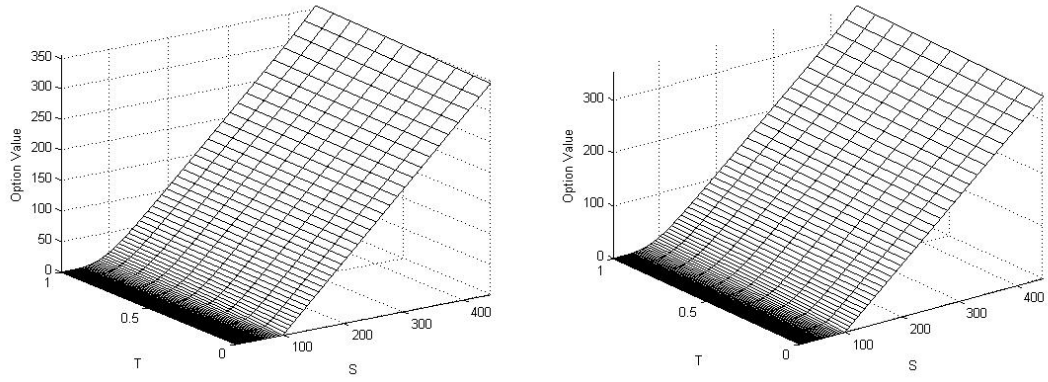


(a) (0,2)-Padè scheme (b) (0,4)-Padè scheme

Figure 9.1: The numerical solution of the European call option obtained by lower and higher order Padè schemes according to the data set in table(9.5) and $x_{\min} = -1.5$ and $x_{\max} = 1.5$.

Similar to the previous graph, we use the same data set and domain to present the evolution profiles for the (0,2)-Padè and (0,4)-Padè schemes in figure(9.2a)

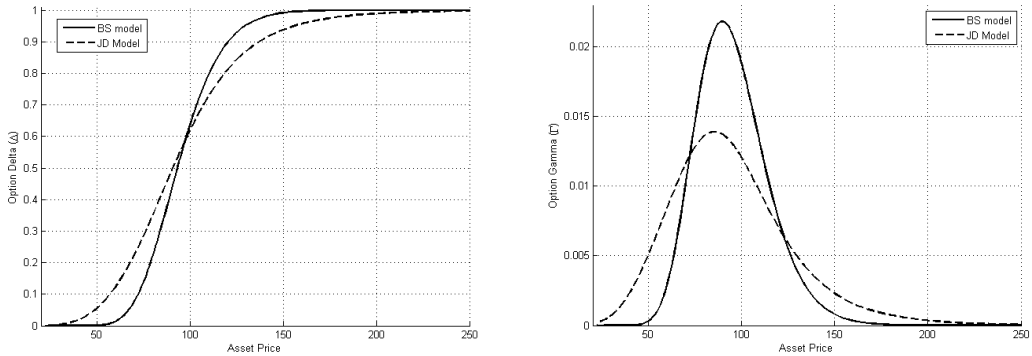
and figure (9.2b) respectively. They both show a similarity with the graph of the solution profile in figure (9.1).



(a) (0,2)-Padè scheme (b) (0,4)-Padè scheme

Figure 9.2: The evolution profile of the European call option obtained by lower and higher order Padè schemes.

To test behaviour of our schemes, we check the Greeks options under the Black-Scholes model , no jump, and the Merton's jump diffusion model with high intensity jump. The graphs in figures (9.3) show that our L-stable schemes are very stable and no spurious oscillations even if the the jump has high intensity.



(a) Delta (b) Gamma

Figure 9.3: Greek options under European call option in Merton's model

The convergence results are computed by multiplying the number of time-steps N by two, starting by 40 nodes and uniformly refined to 640. Whereas, the space step size is fixed by $h = 0.001$. The error is calculated by the difference between the exact and approximated solution at the asset price $S = E$. The exact solution at $T = 0.5$ and $T = 1$ is 10.4219064 and 15.66668082 respectively, where the order of convergence is computed by the formula [39]

$$Order = log_2\left(\frac{error(t_k)}{error(t_{k/2})}\right). \tag{9.2}$$

Table 9.6: Convergence of the L–Stable scheme at $S = E$

	T=0.5		T=1	
Time Steps	*Error*	*Order*	*Error*	*Order*
40	1.2283e-03	–	5.9796e-3	–
80	3.0948e-4	1.98877	1.5008e-3	1.99432
160	7.7761e-05	1.99273	3.7514e-4	2.00022
320	1.9492e-05	1.99615	9.3706e-5	2.00122
640	4.8372e-06	2.0106	2.3296e-05	2.00803

The results in table(9.6) shows the desired second order of convergence in time the L–Stable scheme at short and long maturities.

9.2.2 European Call option under Kou's model

For Kou's jump diffusion model, we consider the parameters given in table (9.1) and the truncated domain $\Omega = [-6, 6]$. We compare our scheme with the second order backward differentiation formula (BDF2) given in a well-known literature, [6]. Table (9.7) illustrates the results of the comparison with respect to error at asset price $S = E$ whose analytical solution is 0.04267.

Table 9.7: European call option under Kou's model

		L–Stable scheme			BDF2		
M	N	Price	Error	Order	Price	Error	Order
65	10	0.026923	1.57469972e-02	–	0.024380	1.82900000e-02	–
129	20	0.036869	5.80130313e-03	1.44063	0.034070	8.60000000e-03	1.08865
257	40	0.041394	1.27590920e-03	2.18485	0.040860	1.81000000e-03	2.24835
513	80	0.042347	3.23055884e-04	1.98167	0.042400	2.70000000e-04	2.74496

From the results in Table(9.7) we can say that our approximated solution obtained by L–Stable scheme is getting closer to the exact solution faster than BDF2 method specially when the step size is smaller. Furthermore, the order of convergence gets also closer to 2 in the Padè scheme faster than in the FDBK2.

9.2.3 European Put Option Under Merton's JDM

We consider the domain $\Omega = [-2, 2]$ and the parameters given in Table(9.3) to test the effectiveness of the (0,2)- Padè scheme in European put option under

Merton's JDM. Figure (9.4) demonstrates the graph of the European put option obtained by the 100 nodes and 20 time-step and the data set in table (9.3).

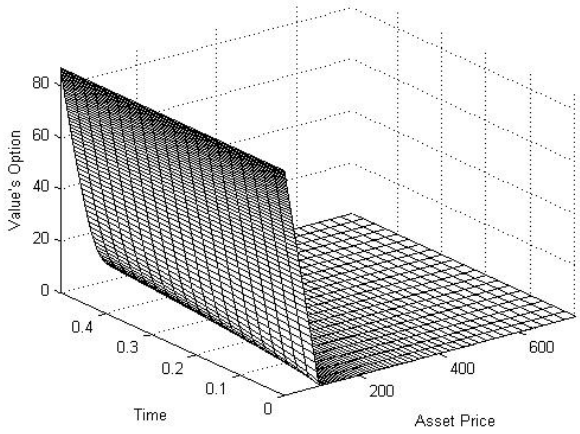


Figure 9.4: Time evolution profile of the European put option

We compute the price of the European put option under Merton's JDM at the underling asset price $S = E$ using the (0,2)-Padè scheme with the same parameters in Table (9.3). The exact solution at $S = 100$ is 15.034989, which is computed by the analytical formula [2] and the error is calculated by equation (9.1).

In this experiment we compare the the efficiency and accuracy of (0,2)-Padè scheme when using two different spatial discretization methods as well as two different integral approximation methods, that are the central finite difference method with the composite trapezoidal rules and the Chebyshiv spectral method with the Clenshaw-Curtis Quadrature rules.

Table 9.8: Finite difference method vs Chebychev spectral method

Steps	Central FD Method			Spectral Method		
M=N	Value at E	Error	CPU	Value at E	Error	CPU
40	14.84558872	1.89400e-01	0.040	15.03228280	2.70620e-03	0.041
80	14.98433811	5.06509e-02	0.100	15.03467073	3.18274e-04	0.121
160	15.02087991	1.41091e-02	0.120	15.03467073	3.18274e-04	0.184
320	15.03080592	4.18308e-03	0.418	15.03497510	1.39022e-05	1.241
640	15.03362563	1.36337e-03	1.659	15.03498484	4.16053e-06	9.471
1280	15.03449027	4.98731e-04	7.661	15.03498720	1.79765e-06	73.595

It can be observed from Table(9.8) that the Chebychev spectral method is approaching the exact solution faster than the finite difference method. However, the time of processing is much higher than of the FD method. This is due to the dense matrix A is not Toeplitz matrix when using spectral method.

9.3 Pricing American Options

In this section we examine the efficiency and accuracy of our schemes in more complex options which are the American options under Merton's and Kou's model. Unlike European options, there are not an analytical solutions for American option so that all the error of all the examined experiments in this section will be

calculated by taking the difference of two successive levels of refinements. *i.e.*

$$Error = |v_{2N}(x,t) - v_N(x,t)| \tag{9.3}$$

9.3.1 American Call Option Under Merton's model

We consider the truncated domain $\Omega = [-2.2, 2.2]$, with the parameters given in table(9.9) and $\epsilon = 0.01$. [42]

Table 9.9: Parameters for American call option.

E	σ	δ	r	λ	μ	T
100	0.15	0.25	0.04	1	0	1

To experimentally prove the order of convergence in space, we shall use the second order central finite difference method and the Chebychev spectral method to discretize the domain Ω in space. Regarding to the integral term (jump part), we use composite trapezoidal rule when using finite difference method. Whereas, the Clenshaw-Curtis Quadrature is used to approximate the integral term when using the spectral method.

The results in table(9.10) are obtained by fixing the time-steps $N = 100$, and compute the American call option by the fourth order (0,4)-Padè scheme at the asset price $S = E = 100$. The error is calculated by equation (9.3), but with respect to space not time, and the order represents the order of convergence computed by the formula(9.2).

Table 9.10: Order of convergence in space of FD and spectral methods

	Central FD Method			Spectral method		
M	Value	Error	Order	Value	Error	Order
40	12.49678969	–	–	12.71511194	–	–
80	12.66399720	1.67208e-01	–	12.71794459	2.83265e-03	–
160	12.70438611	4.03889e-02	2.04960864	12.71811953	1.74943e-04	4.01719043
320	12.71462823	1.02421e-02	1.97944401	12.71813047	1.09423e-05	3.99889477
640	12.71723026	2.60203e-03	1.97680797	12.71813116	6.84221e-07	3.99931642

From table (9.10), it is noticed that the second and fourth order convergence are achieved when using the second order central finite difference method and spectral method respectively in the spatial discretizations.

9.3.2 American Put Option Under Merton's Model

In this experiment we shall test the convergence of the (0,2)-Padè scheme in time when using finite difference and spectral methods. The scheme is tested by pricing the American put option under Merton's JDM in the domain $\Omega = [-1.4, 1.4]$, and the parameters in table (9.11) and $\epsilon = 0.01$.

Table 9.11: Data set for American put option experiment .

E	σ	δ	r	λ	μ	T
100	0.15	0.3	0.03	1	0	0.5

The graph of the numerical solution of American put option under Merton's

model is given in figure(9.5). We also present the behaviour of the Delta and Gamma options in figure (9.6) under American put option in Merton's model. All the figures are presented in the parameters given in table (9.11) and the domain $\Omega = [-1.4, 1.4]$, where N=80 and M=512.

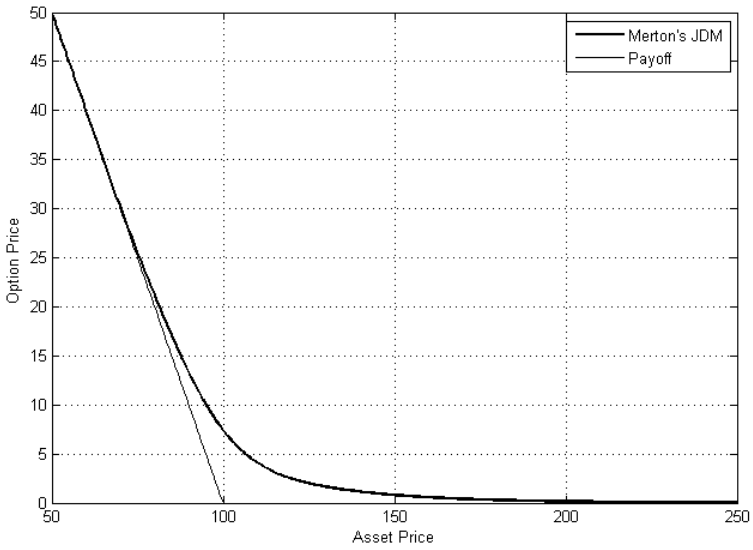
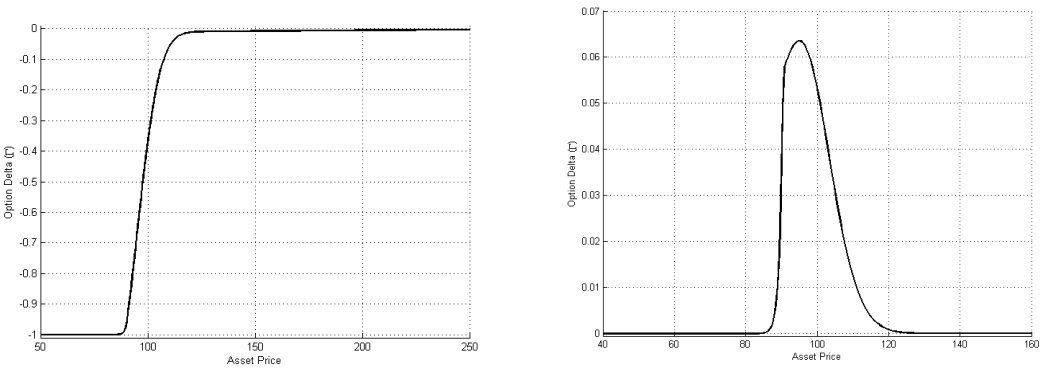


Figure 9.5: Numerical solution of American put option under Merton's JDM



(a) Delta (b) Gamma

Figure 9.6: Greek options under American put option in Merton's model.

In table (9.12), we observe Order of convergence in time of our lower order

L-stable scheme is achieved when using the finite difference method or the spectral method as a spatial discretization. The results in table (9.12) is calculated under the parameters given in table (9.11) and the domain $\Omega = [-1.4, 1.4]$.

Table 9.12: Order of convergence in time at fixed $M = 1000$.

	Central FD Method			Spectral method		
N	Value	Error	Order	Value	Error	Order
40	7.38654964	–	–	7.38681646	–	–
80	7.38682832	2.78683e-04	–	7.38709526	2.78793e-04	–
160	7.38689937	7.10474e-05	1.97177005	7.38716628	7.10234e-05	1.97282795
320	7.38691730	1.79313e-05	1.98629894	7.38718421	1.79321e-05	1.98574951
640	7.38692181	4.50926e-06	1.99152175	7.38718872	4.50701e-06	1.99230254

9.3.3 American Put Option Under Kou's Model

In this experiment we shall test the convergence of the (0,2)-Padè scheme in time when pricing the American put option under Kou's JDM in the domain $\Omega = [-1.5, 1.5]$, and the parameters [13] given in table (9.13)

Table 9.13: Data set for American put option experiment(2)

E	σ	ρ	r	λ	α_1	α_2	T	ϵ
100	0.15	0.3445	0.05	0.1	3.0465	3.0775	0.25	0.001

The graph of the numerical solution of American put option under Kou's model is given in figure (9.7). We also present the behaviour of the Delta and

Gamma options in figure (9.8) under American put option in Kou's model. All the figures are presented in the parameters given in table (9.13) and the domain $\Omega = [-1.5, 1.5]$, where N=80 and M=512.

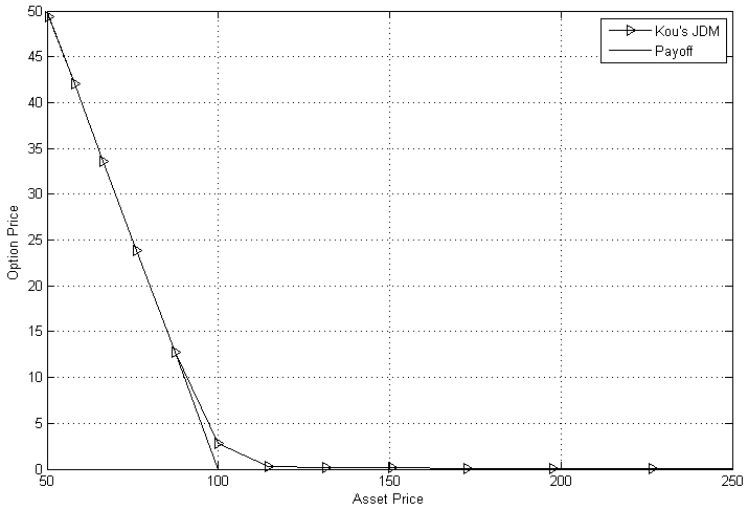
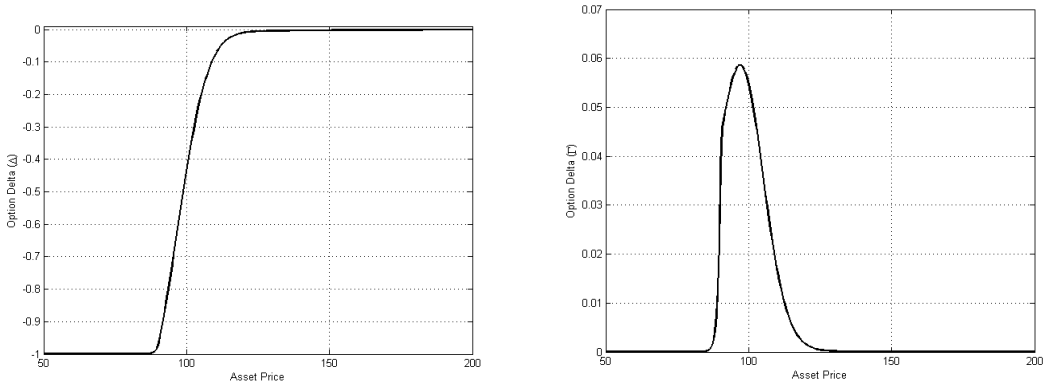


Figure 9.7: Numerical solution of American put option under Kou's model



(a) Delta (b) Gamma

Figure 9.8: Greek options under American put option in Kou's model.

In table (9.14), we observe the order of convergence in time of the L-stable (0,2)-Padè scheme is achieved when we use the finite difference method as a spatial

discretization. The results in table (9.14) is calculated under the parameters given in table (9.13) and the domain $\Omega = [-1.5, 1.5]$.

Table 9.14: Order of convergence in time at fixed $M = 1000$ for Kou's model

N	Value	Error	Ratio	Order
40	3.210517	–	–	–
80	3.210603	8.59891265e-05	–	–
160	3.210625	2.20620565e-05	3.89760	1.96258696
320	3.210630	5.59474814e-06	3.94335	1.97942219
640	3.210632	1.41029229e-06	3.96708	1.98807898

9.3.4 Butterfly Spread call Option

We consider the price of the European and American butterfly spread call option under Merton's JDM obtained by (0,2)-Padè scheme. In this experiment we consider the truncated domain $\Omega = [-2, 2]$ with the parameters given in Table(9.3). Figures (9.9) represents the graph of the European and American butterfly call option compared with the payoff function of same option. It it easy to observe that how far is the American option's value from the European one.

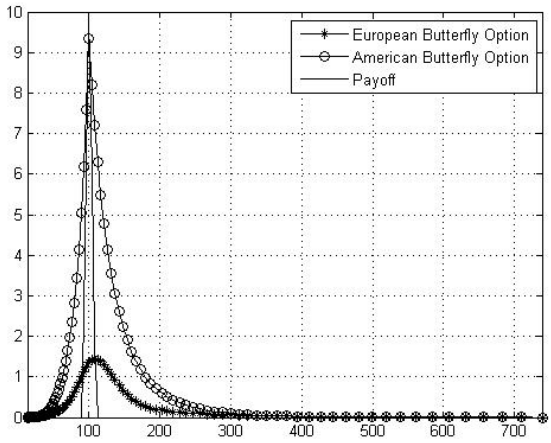


Figure 9.9: Numerical solution of American & European butterfly options

Another view of the difference between the two options can be seen in Figure (9.10) and (9.11) which represent the time evolution profile for European and American butterfly call option respectively. All the three figures are obtained by (0,2)-Padè scheme with 100 nodes and 50 time-step.

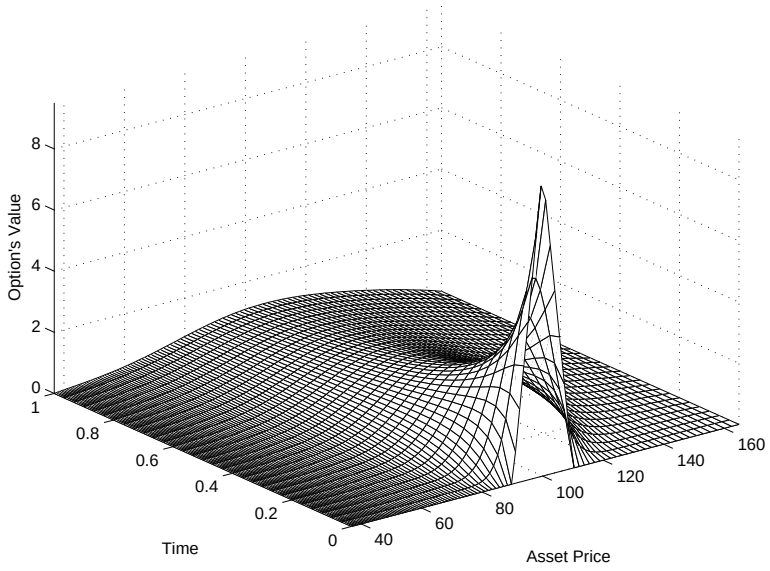


Figure 9.10: Time evolution of the European butterfly spread call option.

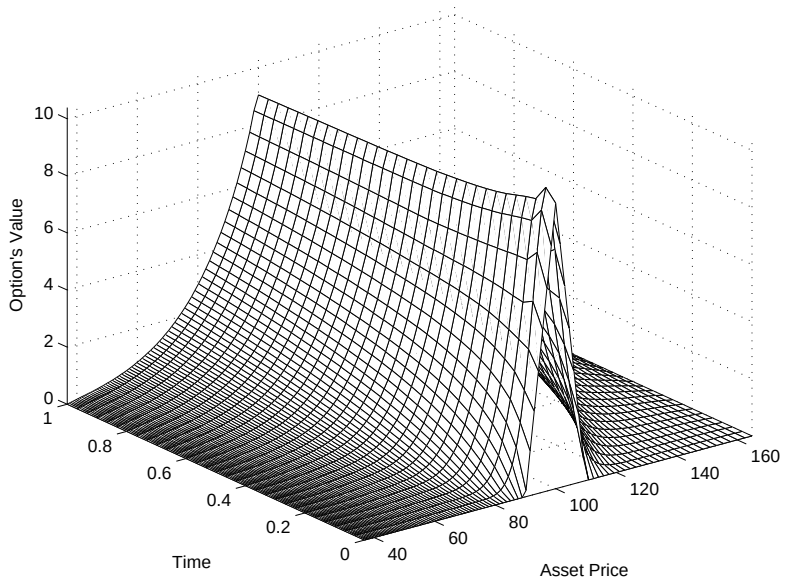


Figure 9.11: Time evolution of the American Butterfly spread call option.

The results in Tabel(9.15) present the values of the European butterfly spread option at the asset price $S = E = 100$, obtained by (0,2)-Padè scheme and (0,4)-Padè scheme with fixed space-steps $M = 1500$ nodes at every refinement.under the data set given in Table (9.3) with the domain $\Omega = [-2, 2]$.

Table 9.15: Order of convergence in time under butterfly option

	(0,2)-Padè scheme			(0,4)-Padè scheme		
N	Value	Error	Order	Value	Error	Order
20	1.842771	–	–	1.841614	–	–
40	1.841917	8.536439e-04	–	1.841612	1.771131e-06	–
80	1.84169	2.266664e-04	1.913064	1.841612	1.235624e-07	3.841359
160	1.841632	5.857267e-05	1.952271	1.841612	8.157522e-09	3.920965
320	1.841617	1.490047e-05	1.97487	1.841612	5.097511e-10	4.000266

It can be observed from Table (9.15) that we have achieved a second and fourth order of convergence corresponding to the (0,2)-Padè and (0,4)-Padè schemes respectively.

CHAPTER 10

CONCLUSION

In this thesis we have developed L-stable numerical methods to obtain numerically the solution for pricing American and European options under Merton's and Kou's jump diffusion models. These models are extensions of the well-known Black-Scholes model. The models are mathematically formulated by semilinear partial integro-differential equations. Our numerical methods are based on Padè approximation in the partial fraction form. These methods are divided into two schemes, lower and higher numerical schemes.

We have used second order central finite difference method as a spatial discretization methods in space. Then we used the Composite Trapezoidal Rule to explicitly approximate the integral part. Using method of line (MOL) we end up with a semidiscrete system of ODEs. Using the Duhamel principle we have found the exact solution of the semi-discrete system. Finally we used (0,2)-Padè schemes which base on the exponential time differencing schemes combined with

Runge-Kutta methods. Where the matrix exponential function is approximated by the (0,2)-Padè approximation. Furthermore, the FFT algorithm has been used as a fast matrix-vector multiplication solver. All these techniques yield an efficient and stable numerical method whose order of convergence is two in space as well as in time.

To get a higher and faster accuracy, we also have used higher order methods. We have used Chebyshiv spectral methods to discretize the space domain, then the Clenshaw-Curtis Quadrature was used to approximate the integral part. Then finally, we have used a fourth order (0,4)-Padè scheme whose construction is like of the (0,2)-Padè scheme. This numerical method also achieves the fourth order of convergence in space as well as in time.

REFERENCES

[1] F. Black and M. Scholes, "The pricing of options and corporate liabilities," *The journal of political economy*, pp. 637–654, 1973.

[2] R. C. Merton, "Option pricing when underlying stock returns are discontinuous," *Journal of financial economics*, vol. 3, no. 1-2, pp. 125–144, 1976.

[3] S. G. Kou, "A jump-diffusion model for option pricing," *Management science*, vol. 48, no. 8, pp. 1086–1101, 2002.

[4] L. Andersen and J. Andreasen, "Jump-diffusion processes: Volatility smile fitting and numerical methods for option pricing," *Review of Derivatives Research*, vol. 4, no. 3, pp. 231–262, 2000.

[5] K. I. Amin, "Jump diffusion option valuation in discrete time," *The journal of finance*, vol. 48, no. 5, pp. 1833–1863, 1993.

[6] A. Almendral and C. W. Oosterlee, "Numerical valuation of options with jumps in the underlying," *Applied Numerical Mathematics*, vol. 53, no. 1, pp. 1–18, 2005.

[7] R. Cont and E. Voltchkova, "A finite difference scheme for option pricing in jump diffusion and exponential lévy models," *SIAM Journal on Numerical Analysis*, vol. 43, no. 4, pp. 1596–1626, 2005.

[8] Y. dHalluin, P. A. Forsyth, and G. Labahn, "A penalty method for american options with jump diffusion processes," *Numerische Mathematik*, vol. 97, no. 2, pp. 321–352, 2004.

[9] D. Tangman, A. Gopaul, and M. Bhuruth, "Exponential time integration and chebychev discretisation schemes for fast pricing of options," *Applied Numerical Mathematics*, vol. 58, no. 9, pp. 1309–1319, 2008.

[10] S. Ikonen and J. Toivanen, "Operator splitting methods for american option pricing," *Applied mathematics letters*, vol. 17, no. 7, pp. 809–814, 2004.

[11] Y. Kwon and Y. Lee, "A second-order finite difference method for option pricing under jump-diffusion models," *SIAM Journal on Numerical Analysis*, vol. 49, no. 6, pp. 2598–2617, 2011.

[12] S. Salmi and J. Toivanen, 'IMEX schemes for pricing options under jump–diffusion models," *Applied Numerical Mathematics*, vol. 84, pp. 33–45, 2014.

[13] M. K. Kadalbajoo, L. P. Tripathi, and A. Kumar, "Second order accurate IMEX methods for option pricing under Merton and Kou jump-diffusion models," *Journal of Scientific Computing*, vol. 65, no. 3, pp. 979–1024, 2015.

[14] M. A. Gondal, "Option valuation in jump-diffusion models using the exponential runge-kutta methods," *World Applied Sciences Journal*, vol. 13, no. 11, pp. 2396–2404, 2011.

[15] S. T. Lee, H.-K. Pang, and H.-W. Sun, "Shift-invert arnoldi approximation to the toeplitz matrix exponential," *SIAM Journal on Scientific Computing*, vol. 32, no. 2, pp. 774–792, 2010.

[16] A. Khaliq, B. Wade, M. Yousuf, and J. Vigo-Aguiar, "High order smoothing schemes for inhomogeneous parabolic problems with applications in option pricing," *Numerical Methods for Partial Differential Equations*, vol. 23, no. 5, pp. 1249–1276, 2007.

[17] R. Zvan, P. Forsyth, and K. Vetzal, "Penalty methods for american options with stochastic volatility," *Journal of Computational and Applied Mathematics*, vol. 91, no. 2, pp. 199–218, 1998.

[18] S. T. Lee, X. Liu, and H.-W. Sun, "Fast exponential time integration scheme for option pricing with jumps," *Numerical Linear Algebra with Applications*, vol. 19, no. 1, pp. 87–101, 2012.

[19] C. R. Vogel, *Computational methods for inverse problems*, vol. 23. Siam, 2002.

[20] S. Cox and P. Matthews, "Exponential time differencing for stiff systems," *Journal of Computational Physics*, vol. 176, no. 2, pp. 430–455, 2002.

[21] A.-K. Kassam and L. N. Trefethen, "Fourth-order time-stepping for stiff pdes," *SIAM Journal on Scientific Computing*, vol. 26, no. 4, pp. 1214–1233, 2005.

[22] A. Khaliq, J. Martin-Vaquero, B. Wade, and M. Yousuf, "Smoothing schemes for reaction-diffusion systems with nonsmooth data," *Journal of Computational and Applied Mathematics*, vol. 223, no. 1, pp. 374–386, 2009.

[23] M. Yousuf, "On the class of high order time stepping schemes based on Padè approximations for the numerical solution of burgers equation," *Applied Mathematics and Computation*, vol. 205, no. 1, pp. 442–453, 2008.

[24] M. Yousuf, A. Khaliq, and B. Kleefeld, "The numerical approximation of nonlinear Black-Scholes model for exotic path-dependent american options with transaction cost," *International Journal of Computer Mathematics*, vol. 89, no. 9, pp. 1239–1254, 2012.

[25] E. Gallopoulos and Y. Saad, "On the parallel solution of parabolic equations," in *Proceedings of the 3rd international conference on Supercomputing*, pp. 17–28, ACM, 1989.

[26] A. Khaliq, E. Twizell, and D. Voss, "On parallel algorithms for semidiscretized parabolic partial differential equations based on subdiagonal padé approximations," *Numerical Methods for Partial Differential Equations*, vol. 9, no. 2, pp. 107–116, 1993.

[27] J. H. Mathews and K. D. Fink, *Numerical methods using MATLAB*, vol. 31. Prentice hall Upper Saddle River, NJ, 1999.

[28] G. Dahlquist and Å. Björck, "Numerical methods in scientific computing, volume i," *SIAM, Philadelphia*, vol. 55, 2007.

[29] T. P. Timsina, "Sensitivities in option pricing models," 2007.

[30] J. F. Epperson, *An introduction to numerical methods and analysis*. John Wiley & Sons, 2013.

[31] G. D. Smith, *Numerical solution of partial differential equations: finite difference methods*. Oxford university press, 1985.

[32] C. Canuto, M. Y. Hussaini, A. M. Quarteroni, A. Thomas Jr, *et al.*, *Spectral methods in fluid dynamics*. Springer Science & Business Media, 2012.

[33] M. Yousuf, *Smoothing schemes for the inhomogeneous linear and semilinear parabolic problems with nonsmooth data*. PhD thesis, PhD thesis, University of Wisconsin-Milwaukee, 2005.

[34] L. N. Trefethen, *Spectral methods in MATLAB*, vol. 10. Siam, 2000.

[35] L. N. Trefethen, *Finite difference and spectral methods for ordinary and partial differential equations*. Cornell University-Department of Computer Science and Center for Applied Mathematics, 1996.

[36] V. Thomée, *Galerkin finite element methods for parabolic problems*, vol. 1054. Springer, 1984.

[37] H. P. Bhatt and A. Q. Khaliq, "Higher order exponential time differencing scheme for system of coupled nonlinear schrödinger equations," *Applied Mathematics and Computation*, vol. 228, pp. 271–291, 2014.

[38] C. Moler and C. Van Loan, "Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later," *SIAM review*, vol. 45, no. 1, pp. 3–49, 2003.

[39] M. Yousuf, A. Khaliq, and R. Liu, "Pricing american options under multi-state regime switching with an efficient l-stable method," *International Journal of Computer Mathematics*, vol. 92, no. 12, pp. 2530–2550, 2015.

[40] G. Wanner and E. Hairer, *Solving ordinary differential equations II*, vol. 1. Springer-Verlag, Berlin, 1991.

[41] M. Yousuf and A. Khaliq, "An efficient etd method for pricing american options under stochastic volatility with nonsmooth payoffs," *Numerical Methods for Partial Differential Equations*, vol. 29, no. 6, pp. 1864–1880, 2013.

[42] N. Rambeerich, D. Tangman, A. Gopaul, and M. Bhuruth, "Exponential time integration for fast finite element solutions of some financial engineering problems," *Journal of Computational and Applied Mathematics*, vol. 224, no. 2, pp. 668–678, 2009.

[43] E. Pindza, K. Patidar, and E. Ngounda, "Robust spectral method for numerical valuation of european options under Merton's jump-diffusion model,"

Numerical Methods for Partial Differential Equations, vol. 30, no. 4, pp. 1169–1188, 2014.

[44] B. Kleefeld, A. Khaliq, and B. Wade, "An ETD Crank-Nicolson method for reaction-diffusion systems," *Numerical Methods for Partial Differential Equations*, vol. 28, no. 4, pp. 1309–1335, 2012.

[45] G. Beylkin, J. M. Keiser, and L. Vozovoi, "A new class of time discretization schemes for the solution of nonlinear pdes," *Journal of Computational Physics*, vol. 147, no. 2, pp. 362–387, 1998.

[46] D. A. Voss and A.-Q. M. Khaliq, "Time-stepping algorithms for semidiscretized linear parabolic pdes based on rational approximants with distinct real poles," *Advances in Computational Mathematics*, vol. 6, no. 1, pp. 353–363, 1996.

[47] A. Khaliq and B. Wade, "On smoothing of the crank–nicolson scheme for nonhomogeneous parabolic problems," *Journal of Computational Methods in Sciences and Engineering*, vol. 1, no. 1, pp. 107–123, 2001.

[48] B. Wade, A. Khaliq, M. Siddique, and M. Yousuf, "Smoothing with positivity-preserving padé schemes for parabolic problems with nonsmooth data," *Numerical Methods for Partial Differential Equations*, vol. 21, no. 3, pp. 553–573, 2005.

[49] J. H. Bramble, J. E. Pasciak, P. H. Sammon, and V. Thomée, "Incomplete iterations in multistep backward difference methods for parabolic problems with

smooth and nonsmooth data," *Mathematics of computation*, vol. 52, no. 186, pp. 339–367, 1989.

[50] S. M. Serbin, "A scheme for parallelizing certain algorithms for the linear inhomogeneous heat equation," *SIAM journal on scientific and statistical computing*, vol. 13, no. 2, pp. 449–458, 1992.

[51] Y. Yan, "Smoothing properties and approximation of time derivatives for parabolic equations: constant time steps," *IMA journal of numerical analysis*, vol. 23, no. 3, pp. 465–487, 2003.

[52] A. Khaliq, D. Voss, and M. Yousuf, "Pricing exotic options with l-stable padé schemes," *Journal of Banking & Finance*, vol. 31, no. 11, pp. 3438–3461, 2007.

[53] G. Dahlquist and Å. Björck, "Numerical methods in scientific computing, volume i," *SIAM, Philadelphia*, vol. 55, 2007.

[54] L. N. Trefethen, "Is Gauss quadrature better than Clenshaw-Curtis?," *SIAM review*, vol. 50, no. 1, pp. 67–87, 2008.

[55] J. Waldvogel, "Fast construction of the fejér and clenshaw–curtis quadrature rules," *BIT Numerical Mathematics*, vol. 46, no. 1, pp. 195–202, 2006.

[56] P. Wilmott, S. Howison, and J. Dewynne, *The mathematics of financial derivatives: a student introduction*. Cambridge University Press, 1995.

[57] D. C. Lesmana, *Numerical Methods for Nonlinear Partial Differential Equations and Inequalities Arising from Option Valuation under Transaction Costs*. PhD thesis, The University of Western Australia, 2014.

Vitae

- Name: SALAH ABDO MURSHED ALRABEEI
- Nationality: YEMENI
- Date of Birth: 01/01/1989
- Email: *salah.arrabyee@gmail.com*
- Permenant Address: Sana'a St., Dhamar, Yemen